COMPUTATIONAL PROPERTIES OF REGULAR PROCESSOR NETWORKS

A THESIS

Presented to

The Faculty of the Division of Graduate Studies

By

Renato Ermann

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

in the School of Information

and Computer Science

Georgia Institute of Technology

June, 1977

COMPUTATIONAL PROPERTIES OF REGULAR

PROCESSOR NETWORKS

Approved:

William I. Grosky, Chairman

Lucio Chiaraviglio

Philip H. Enslow, Jr.

Richard A. DeMillo

Date approved by Chairman SEP 19 1977

ACKNOWLEDGMENTS

I owe a particular debt of gratitude to my principal advisor, Dr. William I. Grosky, who introduced me to this field. Especially, I wish to thank him for the expert guidance, valuable help and strong backing he consistently gave me. Dr. Grosky has taught me much more than is evidenced in this document.

Appreciation is extended to Dr. Lucio Chiaraviglio, Dr. Philip H. Enslow and Dr. Richard A. DeMillo for their advice and criticism as members of the thesis reading committee. They spent many hours with me in shaping and refining the thesis into its present form. Dr. Chiaraviglio aided with the formalization of various concepts and suggested a number of areas for consideration.

Gratitude is likewise expressed to Dr. Edward B. Wagstaff and to Dr. Chee-Yee Chong of the School of Electrical Engineering for their comments on my work and for serving as external readers. Special thanks are due to Dr. Vladimir Slamecka for his support and for providing financial aid in the form of graduate teaching assistantships.

The encouragement, understanding and assistance of my parents and of Eda were invaluable in the completion of this dissertation. I dedicate it to them.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

SUMMARY

Several design problems of large-scale parallel processing systems become considerably easier to resolve if their interconnection patterns are regular. By means of a mathematical device, called a regular processor network (RPN), this dissertation investigates structural properties of such systems.

First, we develop techniques for characterizing the state transition graph of a RPN. In particular, methods are presented for the determination of connectedness and inexistence of unreachable states. A measure of the maximum distance from a given state to another arbitrary state in the transition graph is provided. In addition, necessary structural conditions under which a system can undergo a given sequence of states are established.

Next, we investigate several trade-offs between hardware complexity and computing time in relation to the execution of the class of problems solvable within polynomial-time by nondeterministic Turing machines.

Finally, we derive structural conditions for the existence in a network of decentralized information routing schemes that do not require knowledge of processor sites. The manner in which routing speed of these schemes relates to system topology is dilucidated.

CHAPTER I

INTRODUCTION

1.1. Motivation

The history of computing has been characterized by a constant search for more processing power. This appetite for more and faster processing is likely to continue its growth in the foreseeable future. The need can be satisfied in a number of overlapping ways. As a first approximation, one may distinguish hardware advances, the construction of better types of machines, from software advances, the discovery of better ways of utilizing the machines.

For the past three decades, hardware advances have been due mainly to progress in electronic technology. However, while improvement in density, power consumption and cost of integrated circuit components is likely to continue at roughly the present rate (as has been illustrated by Stone [1]), improvement in logic speed is reaching a limit. This negative fact has been documented, among others, by Ware [2], Winograd [3], and Lamagna and Savage [4].

An alternative method of achieving hardware advances is to design "parallel processing systems." The concept of parallel processing has been used since the dawn of computer history. Twenty-five years ago, logic operations were realized on groups of digits, rather than on one digit at a time. Our interpretation of "parallel processing" will be a broad one: the execution of several activities within a common interval of time. Today, the term at times refers to multiple-computer systems, at other times to multiprocessors and array-type machines (parallel, associative or pipeline), and occasionally to multiprogrammed machines.

With the introduction of the microprocessor, a very inexpensive processor compact enough to be constructed on a single chip, there is an economic motivation for constructing large-scale parallel processing systems from networks of small processors. Component advances have, however, surpassed architectural and algorithmic advances to the extent that it is now possible to build such large-scale networks with tremendous processing power, except that it is not clear what interconnection topology the networks should have and how computations should proceed in them.

There are some classes of problems which may be partitioned into components which are essentially independent of each other. In those cases, it becomes worthwhile to build special purpose parallel processing systems that suit well each specific class of applications. Many examples of such applications are given in surveys by Thurber [5] and Foster [6]. Kuck [7] has demonstrated that most programs will admit some degree of parallel computation, at the bit, operation or algorithm processing levels.

When performing n processes in parallel, Minsky [8] has remarked that it is sometimes the case that a computation is speeded up merely by a factor of $\log_2 n$. Chen [9] among others, has observed also that in a general purpose system, there is a limit to the speedup achievable by unbounded use of parallelism. Thus, the architecture of a general system will have to allow for execution of serial, as well as highly parallel types of tasks. The knowledge of how to perform this dual function in an effective manner is still at an unsatisfactory stage of development. For more discussion on the speedup attainable in general purpose systems, see Kuck et al. [10] and Shore [11].

The design of parallel processing systems presents some challenges not found in single processor systems. Three major factors governing the good use of a given machine are the specification of the order in which computations should be performed, the allocation of loads and resources, and the interaction between system modules. Such control problems can be met more efficiently if the organization of a parallel processing system has a high degree of regularity (see Wulf [59, p. 302]).

The concern of this dissertation is with large parallel processing systems composed of identical processors, interconnected in a regular manner. As a notion of interconnection regularity, it will be specified that all processors in the system have the same finite number of input and output lines. Such a model, called a "regular processor network," is meant to represent both "tightly coupled" systems, such as array processors, and "loosely coupled" systems, such as computer networks. In other words, a regular network could be a single computer with multiple processing units, or several interconnected separate computers. A single operating system could control the entire network, or each of the several computers may have its own largely autonomous operating system. Also, the level at which load and resource sharing take place in the network, is immaterial to the model.

This model is an extension of the "tessellation automaton" of Yamada and Amoroso [12], inspired by von Neumann's cellular automaton [13]. The underlying directed graph of a tessellation automaton is defined in terms of Cartesian coordinates; only those networks for which the direction and orientation of all input lines are the same throughout the Euclidean space can be represented by it. The regular processor network shall be utilized as a tool to study the influence of the interconnection topology of a parallel processing system in its behavior. The notion of interconnection regularity used, is the most general one which preserves the assumption that all processors and connection terminals be identical.

Attempts to evaluate the organization of a system may be done roughly from three points of view: system theoretic, individual problem oriented and statistical. The first approach will be taken in this dissertation. In this view of system structure, the intervening interactions and parameters are considered without respect to their relative importance in a problem environment.

Another reason for the present study are the benefits of a parallel processing system with a regular organization: lower hardware and software development costs, increased reliability and a potential for incremental expansion. The reduction in costs is a clear consequence of resulting economies of scale. The regularity of the system permits the use of production techniques which are not feasible otherwise.

On purely intuitive grounds, it seems that a system composed of a number n of largely independent identical processors, should be more reliable than a system without such redundancy. The advocates of parallel systems argue that if dynamic allocation of tasks is performed, the loss of a processor would merely result in the loss of roughly 1/n of the processing power of the system. They call this behavior graceful degradation. The complete picture, of course, is that the interconnection of the processors adds to the complexity of the network. Regularity in a system's organization can make the corresponding control software (which coordinates the use of resources) simpler and easier to test, since a modular scheme, such as decentralized anonymous treatment of all processors, may be designed. More specifically, decomposition of programs, synchronization of both data and control, interprocessor communication and the mechanisms of resolving contention for shared resources, all become simpler. Good discussions of this subject are given by Enslow [14], Farber [15] and Searle and Freberg [16]. The interprocessor communication problem is studied in detail in Chapter IV.

The ease with which the power of a system can be altered, is sometimes called flexibility. Intuitively, it is clear that a highly flexible system would have to be regularly organized. A properly designed regular processor network may be expanded or contracted in real-time by simply "adding or eliminating processors", a feature that is of particular importance to many applications.

On a more speculative level, an analogy can be made between a large regular network composed of identical small processors, and some biological entities with a similar structure. In fact, von Neumann's work on cellular automata [13] was partly inspired by this analogy. The nervous system is a prime example of a biological system that consists of individually unreliable elements, and yet exhibits a stable and reliable behavior. It can tolerate destruction or removal of large numbers of cells and still continue to function adequately. Such a graceful degradation follows from the interchangeability among neurons present in the brain. This property of the brain may be thought to be analogous to performing dynamic allocation of tasks by a regular processor network.

Another important concept inspired by nature, is that of a system organization which treats processors as an anonymous pool of resources. The routing of information between processors in such a system is determined by a decentralized control mechanism. Interprocessor communication is content-based, that is to say, messages (consisting of both control information and data) are of the "to-whom-it-may-concern" type. No record on the identity of message, senders and receivers is normally kept. Optimization in the whole system is achieved by coordinated local optimization at each processor, rather than by prescriptions emanating from a central mechanism. Since local events are sometimes not predictable, and it is too costly and time consuming to submit them for central judgment, general purpose large-scale systems with centralized control schemes are inefficient. Nature presents us with examples of biologic entities capable of remarkable behavior resulting from the combined activity of elementary autonomous individuals. Goals achievable by some of these entities composed of basic individuals (e.g., ants, cells), are well beyond the added uncoordinated capabilities of the individuals. Decentralized system organizations have been advocated by Farber [15], Chen [9] and Tsetlin [17]. They will be discussed further in Chapter IV.

1.2. Historic Remarks

As we have said, our concern will be with regular networks, composed of a large number of identical processing elements (RPN's).

One of the earlier systems of this type is the Holland machine [18]. Holland's design is a two-dimensional rectangular array of identical "modules," each of which contains a storage register, eight auxiliary registers and associated circuitry. Every module can communicate directly with the four modules nearest to it. Time is quantized and occurs in discrete steps. At each instant, a module is either in an active or an inactive state. Three phases can be distinguished in the operation of this machine: input, path building and execution. During an input phase, a module's storage register is set to an initial value. During a path building phase, an active module determines the location of the storage register upon which its instruction is to operate (the operand), by "opening a path" to that operand. During the execution phase, the active module interprets and executes the instruction stored in its storage register.

The Holland machine has two severe disadvantages that prevent it from being a practical machine. It suffers from path interference, and it is very difficult to program. Comfort [19] has simplified and modified the Holland design, partly alleviating these problems.

One of the early special purpose parallel machines, whose processing section may be considered a RPN, is the Unger machine [20]. This machine has been designed to perform pattern-recognition processing, specifically processing of lines. Line thinning, doubling, extending, and center determination are typical functions that could be accomplished with the machine. The main functional units of the system are the master control and the rectangular array of processors. The master control can issue a parallel command to all processors. A parallel processing system with an organization of this type is usually called an array processor.

The first large-scale array processor designed was SOLOMON I [21]. This design later developed to become ILLIAC IV [22], the largest computer currently operating. Squire and Palais [23] have proposed a machine embedded in a multidimensional cube. The construction of such an array processor, HYPERCUBE [24], has been attempted. It is embedded in a 4-dimensional cube and each processor can communicate directly with its eight nearest processors. The machine is built with microprocessors as processing elements; currently, up to 256 processors are provided.

Although not strictly a RPN because of slight differences among its element minicomputers, the Distributed Computing System [15] (DCS) is worth mentioning for its decentralized organization, which attempts to apply the concept of anonymous treatment of all processors, a notion discussed in the previous section. The topology of the network is a ring and it consists presently of five minicomputers, interconnected by transmission lines. The goals of the DCS design include both load sharing and the sharing of unique resources. A specific description of its control features is given in Chapter IV.

A more detailed historic discussion on designs of parallel processing systems is made unnecessary by the existence of the thorough surveys by Enslow [14], Thurber [5] and Foster [24].

Von Neumann's 29 state cellular automaton [13] was the first RPN designed. Moore [25] later has introduced a more general class of cellular automata, which includes von Neumann's system. He defines a tessellation structure as a system consisting of a "two-dimensional Euclidean space, a set of allowable states, a quiescent state and rules for transition between states." He calls a function which associates a state with each cell, a "configuration." A configuration which can not be produced from some other prior configuration by means of the transition rules is called a "Garden-of-Eden" configuration. He establishes a sufficient condition for the existence of such a configuration in a special tessellation structure in which every cell i is directly connected to the eight other cells such that their coordinates differ by at most one from the coordinates of cell i. Myhill [26] has reformulated Moore's condition and showed that it is necessary as well as sufficient. An analogous condition which applies to arbitrary RPN's will be presented in Chapter II.

More recently, Yamada and Amoroso [27, 28, 29] have explored "tessellation automata," embedded in a multidimensional Euclidean space. Starting with a one-dimensional tessellation automaton with only two states, they have investigated the reachability of an arbitrary state configuration from a specified "primitive pattern." Noting the formal similarity of state pattern generation in tessellation automata and theorem derivations in an axiomatic system, Yamada and Amoroso have given it the name "completeness." In this analogy, the primitive pattern may be considered the axiom, and the configurations ensuing from it the theorems. Yamada and Amoroso [27], and Maruoka and Kimura [29] have settled the completeness problem for a one-dimensional tessellation automaton. However, little is known at present concerning the solution of this problem for higher dimensional tessellation automata [27]. Yaku [30] has proved the inexistence of an algorithm which, for any $d \geq 2$, will determine whether an arbitrary configuration restricted to having all but finitely many quiescent states (called a "finite configuration") is Garden-of-Eden in an arbitrary d-dimensional tessellation automaton. A new algorithm which will decide whether an arbitrary configuration defined over a finite number of cells (called a finite partial configuration) is Garden-of-Eden in an arbitrary RPN, will be given in Chapter II of this dissertation.

In a tessellation automaton, the same local transition function is applied to all cells. The global map determined by a local transition function acting simultaneously on all cells is called a "parallel" transformation. Let τ be a parallel transformation, and let $\bar{\tau}$ be τ restricted to the finite configurations. Richardson [31] has proved that if τ is one-to-one, then $\bar{\tau}$ is onto and that if $\bar{\tau}$ is one-to-one, then τ is onto. Ostrand [32] has studied tessellation automata with a view to finding classes of transformations which preserve certain properties of configurations, such as finiteness, palindromes and periodicity.

Yamada and Amoroso [28] have investigated structural and behavioral homomorphisms in tessellation automata. More specifically, they show that any tessellation automaton can be simulated by another tessellation automaton of the same dimension, whose interconnection structure (defined as H_1 in the next section) is similar to that used by von Neumann and Holland. Every reduction in connection complexity is accompanied by an increase in the size of the state alphabet. Speedups by an arbitrary integer factor are attained by appropriately enlarging the state set. These tradeoffs are pertinent to semiconductor technology since fabrication techniques place a premium on simplicity of connections, but the reduction techniques used are not necessarily optimal.

Among others, Herman and Rozenberg [33] have introduced a family of languages that attempt to describe the development of cellular organisms in a manner that takes morphologic, genetic and physiologic observations into account. These constructs, usually called Lindenmayer systems, consist of a set of transition rules with or without cellular interaction, a set of state symbols and an axiom (the starting state configuration). A string in the language is generated by simultaneous application of all transition rules. Thus, Lindenmayer systems with cellular interaction, and one-dimensional tessellation automata are syntactically analogous. Generative powers and decision problems of several classes of Lindenmayer systems have been studied by Herman and Rozenberg [33]. In a related vein, Lieblein [34] has investigated generation of patterns by two-dimensional tessellation automata, using an axiomatic approach.

In an attempt to describe the synchronization of growing filaments, the following problem, called the "firing squad problem," has been formulated [33]. Find a two input, two output finite-state machine (called a cell) with three distinguished states (s, i and f, say) and an arbitrary but finite number of additional states, such that a one-dimensional array of n such cells (with the end cells constantly receiving one special input each, which shows them that they are end cells) will have the following properties: (1) If all the cells are in state s, they will remain in state s; (2) if one of the end cells is in state i, while the rest of the cells are in state s, the array will undergo a series of transitions ending up with all the cells being in state f. In addition, no automaton will be in state f prior to all the others being in state f, i.e., all cells must be turned on simultaneously.

Balzer [35] gave a minimum time solution of 2n-2 steps by using only 8 states. Generalizations to the firing squad problem include allowing the initial disturbance i to take place anywhere in the array; permitting different types of finite-state machines in the array, each type having a particular speed of reaction to the initiating signal; allowing a random reconnection of the automata in the array before each step; stipulating a cell to divide up into two or more cells during the synchronization process, provided that it and its two neighbors be in appropriate states; and higher dimensional extensions. All these problems are solved in linear-time by applying the technique of sending out waves from the original disturbance, and subsequently from meeting points of already established waves [33]. It is of interest to note that Smith [36, 37] has made repeated use of solutions to the firing squad problem in his design of one- and two-dimensional tessellation automata which recognize topologic features of finite patterns within linear-time.

Another problem concerning tessellation automata, and motivated by developmental studies also, is the "French flag problem," originated by Wolpert [38] as a model for animal regeneration. Informally speaking, the problem is to devise a mechanism by which a one-dimensional array of identical finite-state machines, initially divided into three sections of states (equally large blue, white and red sections, as is the French flag), and which is cut into arbitrary pieces, will "regenerate" in such a way that each piece again will generate this blue-white-red configuration. Solutions of this problem [33] are based on the traveling waves technique utilized for solving the firing squad problem.

Since RPN's are machines with unbounded parallelism, they are computationally more powerful than those which can modify only a bounded amount of their storage on a single computational step (multitape Turing machines are a typical example of the latter), called bounded-activity machines. For example, Fischer [39] has shown that one-dimensional tessellation automata can be used to generate prime integers in real-time, something bounded-activity machines can not do.

The computational power of nondeterministic d-dimensional iterative arrays, devices first introduced by Cole [40], has been studied by Seiferas [41]. A d-dimensional iterative array is essentially a tessellation automaton with a distinguished cell. Since the machine is nondeterministic, each value of its transition function is a set of states rather than a single state. Seiferas has shown that every language accepted within time n^d by a nondeterministic multihead Turing machine is also accepted within linear-time by a nondeterministic d-dimensional iterative array. Smith [36] has exhibited a context-free language accepted within real-time by a one-dimensional tessellation automaton, which is not accepted within time of order n^2 by any Turing machine, where n is the length of the input string.

1.3. Regular Processor Networks

The basic definitions and concepts involving regular processor networks (RPN's) are presented in this section. The definitions given here generalize the notions of a "tessellation automaton" due to Yamada and Amoroso [12], and of a "modular computer" due to Wagner [42]. The model attempts to describe a system consisting of a "regularly" interconnected collection of identical processing elements. The notion of regularity used, the most general one compatible with the assumed uniformity of the processing elements and of their connection terminals, requires that all points in the network's labeled directed graph must have the same finite number k of distinctly labeled input and output lines. Thus, exactly one input line of each of k labels must be incident with every point in the network. A labeled directed graph so specified is shown to be isomorphic to a quasigroup-graph of some quasigroup, a concept to be introduced in Definition 1.2.

As a more restricted notion of interconnection regularity, it is then required that the underlying directed graph "look" the same when viewed from any of its points. This informal statement is made precise by stipulating that every two points be mutually mapped by an automorphism of the directed graph. The notion is proved to be equivalent to the specification of the network's underlying labeled directed graph as a group-graph of a group.

This algebraic characterization of directed graphs, not only provides a concise tool for describing and generating examples of interconnection structures that exhibit regularity, but also allows a formulation of their properties in algebraic terms. Before presenting the definition of a RPN, some introductory material is necessary. The following graph-theoretical terminology is fundamental.

DEFINITION 1.1

A *directed graph* D, or, more briefly, a *digraph* D consists of a nonempty set of *points* P(D) together with a collection A(D) of ordered pairs of points. Any such pair (p,q) is called an *arc*. We represent D on paper by drawing a directed line from point p to point q, whenever (p,q) is an arc in D. The arc (p,q) goes from p to q and is *incident* with p and q. By definition, there is at most one arc (p,q) incident with p and q. We also say that p is *adjacent to* q and q is *adjacent from* p. Whenever both (p,q) and (q,p) are arcs in D, it is convenient to draw a single undirected line between points p and q, denoting the pair of directionally opposite arcs. The *outdegree* od(p) of a point p is the number of points adjacent from it, and the *indegree* id(p) is the number adjacent to it. If id(p) = od(p)= k, point p is of *degree* k. A digraph is *arc-labeled* when the arcs are assigned labels (or colors).

A *walk* in a digraph is a nonempty alternating sequence of points and arcs, beginning and ending with points, such that each arc goes from the point preceding it, to the point following it. The *length* of a walk is the number of occurrences of arcs in it. A *closed walk* has the same first and last points, and a *spanning walk* contains all the points. A *path* is a walk in which all points are distinct; a *cycle* is a closed walk of length at least one, with all points distinct, except the first and last. A *line* is an infinite path. If there is a path from point p to point q, then q is said to be *reachable from* p, and the *distance*, d(p,q), from p to q is the length of any shortest such path.

A digraph is *connected* if every two points are mutually reachable. Clearly, a digraph is connected if and only if it has a spanning closed walk. A connected digraph is *finitely connected* in case the distance between every two points is finite. Two digraphs are *isomorphic* if there exists a one-to-one onto mapping between their point sets which preserves (directed) adjacency. An *automorphism* of a digraph D is an isomorphism of D with itself. A digraph D is called *quasiregular* of degree k, if for all points p in P(D), od(p) = id(p) = k. If D is connected and of countable degrees, then D has a countable number of points and arcs. Two points p and q of a digraph D are *similar* if for some automorphism α of D, $\alpha(p) = q$. A *fixed point* is not similar to any other point. A digraph is *point-symmetric* if every pair of points are similar. Every point-symmetric digraph is quasiregular, but not conversely. A counterexample illustrating the latter statement is given later in this section. A digraph is k-*arc-colorable* if an assignment of k colors to its arcs can be found, so that no two arcs with a common initial or terminal point have the same color.

A *subgraph* of a digraph D is a digraph having all its points and arcs in D. A *spanning subgraph* of D is a subgraph containing all the points of D. A 1-*factor* of a digraph D is a spanning subgraph of D which is quasiregular of degree 1. We say that D is the *sum* of spanning subgraphs if it is their arc-disjoint union, and such a union is called a *factorization* of D. If D is the sum of 1-factors, D is called 1-*factorable*. Clearly, a 1-factor of a digraph is a union of disjoint cycles and lines. Whenever a quasiregular digraph D is the sum of k 1-factors, then D is k-arc-colorable, since each 1-factor may be assigned a distinct color, and conversely. A more detailed treatment of these concepts is found in Harary's book [43].

Some algebraic terms are defined next.

DEFINITION 1.2

Nonempty set Q, together with a binary operation • defined in Q (often denoted by juxtaposition, i.e., $a_1 \cdot a_2 = a_1 a_2$, for a_1, a_2 in Q), constitutes a *quasigroup* whenever the following two postulates are satisfied:

Postulate 1 (closure): For all a_1, a_2 in Q, $a_1 a_2$ is also an element of Q.

Postulate 2 (cancellation): For all a,b in Q, the equations ax = b and ya = b have uniquely determined solutions x and y.

A quasigroup constitutes a *group* whenever the following additional postulate is satisfied:

Postulate 3 (associativity): For all a_1, a_2, a_3 in Q,

$$a_1(a_2a_3) = (a_1a_2)a_3.$$

As is customary, we shall write Q to denote a quasigroup with set Q and operation •, and tacitly assume the operation is known. Binary operation • is called the *quasigroup operation.*

PROPOSITION 1.1

A group G has the following two properties:

Property 4 (identity): There is an element e in G such that ea = ae = a for all a in G.

Property 5 (inversion): For each a in G, there is an element denoted a^{-1} such that $aa^{-1} = a^{-1}a = e$. Element a^{-1} is called the *inverse* of a.

PROOF (included for completeness)

Identity: Suppose ea = a. Since (ae)a = a(ea) = aa, postulate 2 implies ae = a. That postulate requires e to be unique.

Next, suppose there are b,d in G such that bd = b. By postulate 2, ya = b for some y in G. Then be = yae = ya = b. Therefore, d = e.

Inversion: Suppose ab = e. Then a(ba) = (ab)a = ea = a. Since ae = a, then ba = e.

Finally, assume ab = ba = e and ca = e. Then

$$c = ce = c(ab) = (ca)b = eb = b. \qquad \Delta$$

Two elements a and b in a quasigroup Q *commute* in case ab = ba. A quasigroup is *Abelian* whenever every two elements in Q commute. Let b be an element in a quasigroup Q. We denote $b^1 = b$; if $r > 1$, then $b^r = bb^{r-1}$. If b is an element in a group G we define the *powers* of b as follows: for a positive integer r, b^r is described as above; $b^0 = e$; finally, $b^{-r} = (b^{-1})^r$. A group G is called *cyclic* with *generator* b in case every element g of G may be expressed as $g = b^r$ for some integer r. The group of integers under addition (here the integer zero is the identity element) is an example of a cyclic group; indeed, integer 1 can be used as a generator. A cyclic group is characterized by the *order* of its generator b, the smallest positive positive integer r such that $b^r = e$. If $b^r = e$ implies r=0, we say b is of *infinite order.*

Next, Definitions 1.1 and 1.2 are related by introducing the concept of a quasigroup graph.

DEFINITION 1.3

Let Q be a quasigroup and H a finite subset of set Q. The *quasigroup-graph* $D_{Q,H}$ is the digraph with point set $P(D_{Q,H}) = Q$ and arc set $A(D_{Q,H}) = \{(hx,x) | x\varepsilon Q, h\varepsilon H\}$.

Trivial consequences of the definition are the following. Digraph $D_{Q,H}$ is quasiregular of finite degree equal to the cardinality of H, called k, and is k-arc-colorable. A k-arc-coloring of $D_{Q,H}$ is simply obtained by assigning color h to arc (hx, x), $x\varepsilon Q$, $h\varepsilon H$.

Dorfler [44] recently has shown that the converse of these statements is true also (however, he has considered finite digraphs only). The proof of the converse requires the results of the following Propositions 1.2 and 1.3, that are provided for the infinite case.

PROPOSITION 1.2

The following statements are equivalent for a digraph D of finite degrees.

For every point p of D, id(p) = od(p).

The set of arcs of D can be partitioned into cycles and lines.

D has a 1-factor.

PROOF

It is given in Harary's book [43] for finite digraphs. Ore [45] and Nash-Williams [46] have shown the infinite counterpart. Δ

DEFINITION 1.4

For the remainder, unless otherwise specified, a *digraph* will have a finite or countably infinite point set and all its points will be of finite outdegree and indegree. For any set S of points of digraph D, the *induced subgraph* D_S is the maximal subgraph of D with point set S.

PROPOSITION 1.3

A finitely connected quasiregular digraph D is 1-factorable.

PROOF

It has been first given by Petersen [47]. Since it does not appear in well-known textbooks, we will include it here. Induction is used as a method.

By Proposition 1.2, D has a 1-factor. Suppose D is quasiregular of degree k. We will show that D is the sum of k 1-factors. This is evident for k = 1. Suppose the property is true for every digraph D' of degree smaller than k. Since digraph D admits a 1-factor F, the digraph obtained by removing the arcs of F is quasiregular of degree k-1. By virtue of the induction hypothesis, this new digraph is the sum of k-1 1-factors. Therefore, D is 1-factorable. Δ

DEFINITION 1.5

Let AL be any finite set of symbols, called an *alphabet*; the symbols are called *letters* of the alphabet. Given an alphabet AL, the set AL^{-1} denotes the set of symbols a^{-1}, where a is a letter in AL. A *positive formula* over alphabet AL is defined recursively as follows: every letter a in AL is a positive formula over AL; if f_1 and f_2 are positive formulas over AL, so is $(f_1 f_2)$. For example, if AL = {a,b}, then (ba)(a(ab)) is a positive formula over AL. A *positive word* over alphabet AL is any finite sequence of letters from AL. The *empty word*, denoted by e, is the positive formula consisting of no letters. AL* denotes the set of all positive words over AL including the empty word. A *word* over alphabet AL is a positive word over $AL \cup AL^{-1}$. The number of letters in a formula (a parenthesis is not a letter) or word w, is the *length* of w. An equality r = s, where r and s are positive formulas over alphabet AL is called a *relation* over AL. If s = e, positive formula r is said to be a *relator* over AL.

Let γ be a one-to-one mapping of a set of symbols H into the set of elements of a quasigroup Q. Suppose that every element of Q is expressible in terms of elements in subset $\gamma(H) \subseteq Q$. Then H is called a set of *generating symbols* for Q, and its image in Q under the mapping γ, a set of *generating elements* for Q. Sometimes both generating elements and generating symbols will be referred to as *generators* for Q. If quasigroup operation is mapped to formula juxtaposition, then under mapping γ, every element of Q is expressible by some positive formula over alphabet H. If every equality satisfied by a set of generators H for Q is derivable from a set of relations R over H, then R is called a set of *defining relations* for Q on the set of generators H. We call the pair (H,R) a *presentation* of quasigroup Q and write Q = (H,R). We say a presentation (H,R) is *finitely generated* if set H is finite. It is *finitely related* if set R is finite. If both H and R are finite, (H,R) is said to be *finite*. If quasigroup Q is a group, every relation in R may be represented by an equality w = e, where w is a word over H, since r = s is equivalent to $rs^{-1} = e$.

Given an arbitrary set of symbols H and an arbitrary set of relations R over H, there is a unique quasigroup (up to isomorphism) with presentation (H,R). A proof of this statement is found in the book by Kurosh [48, p. 140]. Each element of the quasigroup defined by a presentation (H,R) corresponds to a class of equivalent positive formulas (or words, if Q is a group) over H, which are derivable from one another by virtue of the relations in R.

The following main theorem extends Dorfler's result.

THEOREM 1.4

If D is a finitely connected quasiregular digraph, then D is isomorphic to $D_{Q,H}$ for some quasigroup Q and some finite subset H of Q.

PROOF

If D is of degree $k \geq 1$, then by proposition 1.3, D is the sum of k 1-factors. Color the arcs of 1-factor j with color p_j, $j=1,..k$, where $p_1,\ldots,p_k$ are arbitrary points in P(D). This is a k-arc-coloring of D. Our aim is to construct a quasigroup Q with set P(D), such that $D = D_{Q,H}$ with $H = \{p_1,\ldots,p_k\}$. We do this by stipulating that $p_j p_i = p_m$, for p_j in H and p_i,p_m in P(D), whenever arc (p_m,p_i) has been assigned color p_j in the above arc-coloring of D. If there is a walk from an arbitrary point p_v in P(D) to a point p_j in H with color sequence $c_1,\ldots,c_u$ in H, then $p_v = c_1(c_2(\ldots.(c_u p_j)..)$. Therefore, since D is finitely connected, every element in P(D) can be represented by a finite positive formula over H. If two distinct walks represented by color sequences $c_1,\ldots,c_r$ and $d_1,\ldots,d_s$ go from point p_v to point p_t, then

$$p_v = c_1(c_2(..(c_r p_t)..) = d_1(d_2(..(d_s p_t)..).$$

Let the set of all these equalities over H be called R. Then the desired quasigroup Q has presentation (H,R).

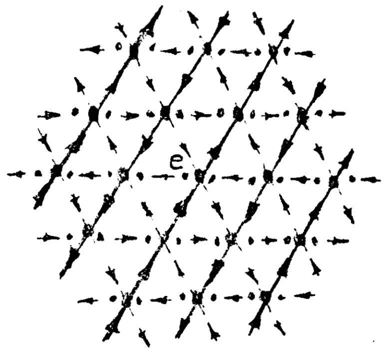
If D has a finite number of points, we may obtain from D a quasigroup operation table for Q, as in the paper by Dorfler [44]. The above stipulation $p_j \cdot p_i = p_m$ whenever (p_m, p_i) has color p_j, specifies the first k rows of the table of Q, with the p_m's as entries. Each of these rows contain every p_m in P(D) exactly once, and each p_m occurs at most once in a column of the rectangle. A k *by* n *Latin rectangle*, with $k \leq n$, has k rows and n columns consisting of integers from 1 to n, with no integers repeated in a row or column. A *Latin square* of order n, is a n by n matrix consisting of integers 1,..,n, in such a way that each integer appears exactly once in each row and in each column. By definition of arc-coloring, the partial operation table stipulated above is a k by n Latin rectangle, where n is the number of points. The task of completing the table is equivalent to the task of augmenting this Latin rectangle into a corresponding Latin square of order n (since cancellation must hold in Q). This augmentation process will sequentially adjoin rows to the Latin rectangle, so that each new rectangle formed is still Latin. The details of this well known process are found, for example, in Liu's book [49]. Δ

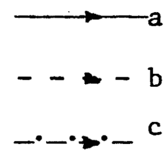
It was asserted above that a quasigroup may be constructed as an algebra whose elements are equivalence classes of positive formulas over a given set of symbols H. This is discussed now in more detail. Two positive formulas are equivalent if it is possible to derive one from the other, by means of a finite number of transformations specified by a set of relations R, and the postulates of a quasigroup, closure and cancellation. Theorem 1.4 stated that every finitely connected quasiregular digraph D may be represented as a quasigroup-graph $D_{Q,H}$ of some quasigroup Q, such that Q=P(D), and set H has a number of elements equal to the degree k of D. A presentation (H,R) for Q is obtained from D by first expressing D as the sum of k 1-factors, k-arc-coloring D with the symbols in H; and finally determining a set of defining relations R over H as follows. Given two points p,q of D, every two distinct walks from p to q imply an equality r=s of positive formulas, r,s spelled by the two walks, as discussed in proof of Theorem 1.4. A set of these equalities, such that none is derivable from the others (and the postulates of a quasigroup), constitutes a set of defining relations R over H. Although each element of Q is a set of equivalent positive formulas, we will denote every element by a positive formula in its class. These concepts are illustrated in Figure 1.1, in which the quasigroup-graph $D_{Q,H}$ associated with a quasigroup Q is exhibited. Each type of line used to represent an arc in $D_{Q,H}$ indicates its color. Note that $D_{Q,H}$ is quasiregular but not point-symmetric, since there are two classes of mutually similar points. Quasigroup Q is specified by a presentation (H,R), obtained from $D_{Q,H}$ as discussed above. Every 1-factor in $D_{Q,H}$ is a collection of lines. Each of the equalities in R corresponds to a path in $D_{Q,H}$ as discussed in Theorem 1.4.

Since a group is an associative quasigroup, the above discussion of quasigroups applies to groups. However, some simplifications do take place. Associativity of a group allows the construction of an algebra whose elements are equivalence classes of words over a generating set H; thus, parentheses can be eliminated.

DEFINITION 1.6

A *regular* digraph is a finitely connected point-symmetric digraph. A *semiwalk* is the undirected equivalent of a walk. More precisely, a semiwalk is a nonempty alternating sequence of points and arcs,





LEGEND: $Q = (H, R)$

$H = \{a,b,c\}$;

$R = \{g(h(gh)) = h(g(hg)) = e,\ f(h(gv_h)) = g(h(fv_h)) = v_h,$

$h(f(h(g(hv_h)))) = h(g(h(f(hv_h)))) = v_h,\ f(f(g(f(hu)))) = u,$

$f(f(g(g(h(hu))))) = u\}$, where f, g, $h \varepsilon H$ are mutually distinct, u, $v \varepsilon H^*$ have an even number of occurrences of each $j \varepsilon H$, except that the number of occurrences of h in v_h is odd, and e is the empty word.

Figure 1.1. A Quasiregular Digraph.

$p_1, a_1, p_2, a_2, \ldots, a_{r-1}, p_r$, where each arc a_i is either (p_i, p_{i+1}) or (p_{i+1}, p_i). A *semipath*, *semicycle*, and so forth, are defined as expected. An *arc-coloring* λ of a regular digraph D is *regular*, in case for every two points p and q of D there exists an automorphism α that preserves λ, and $\alpha(p) = q$.

The following theorem is fundamental. A proof of a weaker result is found in the book by Magnus, Karras and Solitar [50]. They assume given a regular digraph which admits a regular arc-coloring, and show that this digraph is isomorphic to some group-graph. However, they fail to recognize that every regular digraph has some regular arc-coloring. Thus, only the first assumption is necessary.

THEOREM 1.5

If D is a regular digraph, then D is isomorphic to $D_{G,H}$ for some group G and some finite subset of H of G.

PROOF (1) First we show that there exists a regular arc-coloring λ of regular digraph D. By hypothesis, D "looks the same" from every point; an automorphism tells us how to view D from a point q, so that D will appear as it did when viewed from another point p. Formally, if for every two points p,q of D there exists an automorphism α_{pq} such that (i): $\alpha_{pq}(p) = q$ and (ii): $(u,v) \varepsilon A(D)$ if and only if $(\alpha_{pq}(u), \alpha_{pq}(v)) \varepsilon A(D)$, then there is an arc-coloring λ of D for which arc (u,v) is assigned the same color as arc $(\alpha_{pq}(u), \alpha_{pq}(v))$.

D is quasiregular, so by Proposition 1.3, it may be expressed as a sum of 1-factors. We claim that there is a sum such that each 1-factor is point-symmetric, i.e., is the point-disjoint union of mutually isomorphic cycles or lines. The proof consists of two steps.

First, we show that D has at least a 1-factor that is point-symmetric. Choose a minimal subgraph Y of D, that is quasiregular of degree one, and has a point set of cardinality that divides the cardinality of P(D). Such a subgraph Y exists, since by Proposition 1.3, D is 1-factorable. If Y is a spanning subgraph of D, Y is the desired 1-factor. Otherwise, pick a subgraph Y' mutually isomorphic and disjoint to Y, such that a point q of Y' and a point p of Y are adjacent. Since every two points p, q of D are similar, there is a closed walk starting at p, if and only if there is an isomorphic closed walk starting at q. Thus, $Y' = \alpha_{pq}(Y)$, for automorphism α_{pq} specified as above must exist. Define $Z_{S,T} = \{\alpha_{pq}(T)\colon p,q \ \varepsilon\ S\}, S,T \subseteq P(D)$. Now determine $Y'' = Z_{Y,Y'} \cup Z_{Y',Y}; Z_{Y,Y'}$ is the set of subgraphs that "look the same" as Y' when viewed from the points of Y. Next, take Y''' = union of all $Z_{S,T}$, such that S, T are mutually disjoint subgraphs of Y'', isomorphic to Y, and so on. The sequence of subgraphs Y'', Y''',... will converge to a desired point-symmetric 1-factor; otherwise if a point p had not been included in the sequence, this would mean that p was not similar to some adjacent point in the sequence, a contradiction.

Second, we show that D may be expressed as a sum of k 1-factors such that each is point-symmetric, where k is the degree of D. This was just shown for k = 1. Assume it holds for every digraph D' of degree smaller than k. Determine a point-symmetric 1-factor F and remove its arcs; the resulting digraph D'' is point-symmetric of degree k-1. This digraph D'' is either the union of mutually disjoint connected subgraphs, or the union of lines. By virtue of the induction hypothesis, D'' is the sum of k-1 1-factors in the former case, and a point-symmetric 1-factor in the latter case.

Given two arbitrary points p,q of D, we may construct a mapping α_{pq} of P(D) onto itself, such that $\alpha_{pq}(p)=q$, and (u,v) is an arc of point-symmetric 1-factor Z if and only if $(\alpha_{pq}(u), \alpha_{pq}(v))$ is an arc of Z. The mapping is one-to-one and onto, from the definition of 1-factorization, and therefore an automorphism of D that preserves an arc-coloring of the digraph.

Given a point-symmetric 1-factorization of D, there is exactly one such automorphism α_{pq}, for each two points p, q of D. Suppose there existed automorphisms α_{pq} and α'_{pq}, such that $\alpha_{pq} \neq \alpha'_{pq}$. Then $\alpha_{pq}(u) \neq \alpha'_{pq}(u)$, for some point u. Let π be a walk from p to u, and w the sequence of colors of its arcs. Since both automorphisms preserve the given arc-coloring, the sequence of colors of $\alpha_{pq}(\pi)$ and $\alpha'_{pq}(\pi)$ also equals w. These latter two walks both start at point q. Thus, they are identical, since specifying the color sequence of a walk with a given initial point, uniquely determines it (this property readily follows from the definition of an arc-coloring, by induction on the length of the walk, and is true for any quasiregular digraph). Therefore, the endpoints of the two walks, $\alpha_{pq}(u)$ and $\alpha'_{pq}(u)$, are the same, a contradiction. Given a point-symmetric 1-factorization of D, an arc-color-preserving automorphism is uniquely determined.

(2) The construction of a presentation for group G such that D is isomorphic to $D_{G,H}$ may be carried out, of course, as indicated in Theorem 1.4. However, it is convenient to introduce some simplifications. The group is constructed as an algebra whose elements are equivalence classes of words over a set H of k points of D, adjacent to a common point p, where k is the degree of D. A presentation (H,R) for G is determined from D by first expressing the digraph as the sum of k 1-factors, then k-arc-coloring D with the symbols in H, and finally determining a set of defining relations R of words over H, as follows. Every closed walk with p as a starting point that spells a color sequence w, implies an equality w = e, and thus specifies a relator w over H. Since D is a regular digraph, each point in D is associated with the same set of relators. We may represent point p by the empty word e, which corresponds to the identity element of group G.

A semiwalk in D is associated with a word over H, as follows. A semiwalk $p_1,a_1,p_2,\ldots,a_{r-1},p_r$ is represented by word $b_1^{\varepsilon_1}b_2^{\varepsilon_2}\ldots b_{r-1}^{\varepsilon_{r-1}}$, $r \geq 2$, whenever $b_i \varepsilon H$ is the color of arc a_i, i=1,..,r-1, and $\varepsilon_i = 1$ if

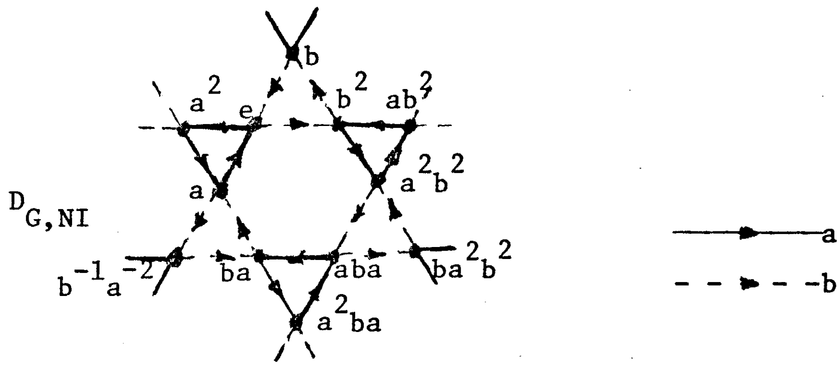
$a_i = (p_i, p_{i+1})$, $\varepsilon_i = -1$ if $a_i = (p_{i+1}, p_i)$. The validity of this correspondence follows from the inversion property of a group. Hence, an element in the associated group G is represented by word w, if and only if there is a semipath of D from point q to point e that spells word w, as specified above. Thus, the relators of G are precisely those words that spell closed semipaths in D.

We need to show that the group-graph of G with presentation (H,R) so constructed, is isomorphic to D. Let α be a mapping from P(D) onto $P(D_{G,H}) = G$ such that $\alpha(p) = w$, whenever there is a semipath from point p to point e that spells word w in G, and $\alpha(e) = e$. That the mapping is one-to-one and onto, follows from our discussion on presentation of quasigroups and the definition of arc-coloring of a digraph. By definition, (v,w) is an arc of $D_{G,H}$ whenever $v = hw$, $h \varepsilon H$. Also, (p,q) is an arc of D whenever there is a semipath from q to e that spells word x, and there is a semipath from p to e that spells hx, $h \varepsilon H$. Thus, (p,q) is an arc of D if and only if $(\alpha(p), \alpha(q))$ is an arc of $D_{G,H}$ and therefore α constitutes an isomorphism between D and $D_{G,H}$. Δ

Part (2) of the proof of Theorem 1.5 is equivalent to the one given by Magnus, Karras and Solitar [7].

The converse of Theorem 1.5 is clearly also valid if H is a set of generators for G, since we may choose the automorphism α_{pq} of $D_{G,H}$ such that $\alpha_{pq}(u) = up^{-1}q$, for all $p,q,u \varepsilon G$. Mapping α_{pq} preserves adjacency and the arc-coloring of $D_{G,H}$, since u=hv is equivalent to $\alpha_{pq}(u) = up^{-1}q = hvp^{-1}q = h\alpha_{pq}(v)$, for $h \varepsilon H$ (by associativity). Thus, every group-graph is point-symmetric.

A group-graph $D_{G,H}$ is shown in Figure 1.2. Note the conciseness of presentation (H,R) of G. Every 1-factor in $D_{G,H}$ is a collection of triangular cycles.



$$G = (H,R)$$

$H = \{a,b\}$ $\qquad\qquad$ $R = \{a^3 = b^3 = (ab)^3 = e\}$

Figure 1.2. A Regular Digraph.

The concept of a so-called regular processor network (RPN) will be provided as a model for an interconnected collection of identical processing elements called cells, represented by finite state machines, each of which has the same finite number k of input terminals and one output terminal. It is assumed that the input lines of each cell are distinctly labeled, and that its output terminal is connected to other k distinct cells. For simplicity, a RPN is specified to operate in a synchronous manner. The state of each cell at time t+1, is given by a local transition function of the states present on the input lines of the cell at time t. The local function applied at a given time instant may not be the same for all the cells.

For convenience, the underlying labeled connection digraph of a

RPN is characterized as a quasigroup-graph of some quasigroup. This formulation not only provides a generally concise way of describing the structure of a digraph (especially if it is highly regular as desired), but also permits a description of its properties in algebraic terms.

By Theorem 1.4, the underlying labeled digraph of a RPN so defined is connected, and the indegree and outdegree of all its points is the same, and equal to a finite integer k. Given an underlying quasigroup-graph $D_{Q,NI}$ with quasigroup Q and finite subset NI of Q, the constraint of connectivity is incorporated by requiring NI to be a set of generators for Q.

DEFINITION 1.7

A *regular processor network* (RPN) is a 4-tuple (A, Q, NI, I) where

A is a finite, nonempty set of states called the *state alphabet;*

Q is a quasigroup, whose underlying set is called the *cellular space*, composed of *cells*;

NI is a finite, nonempty, indexed set of cells $\{h_1,h_2,\ldots,h_k\}$ whose k elements are generators for quasigroup Q; the set NI is referred to as the *neighborhood index* of the RPN; and

I is a nonempty set of mappings from A^k to A called the *admissible local functions.*

The interconnection structure of a RPN is specified by the quasigroup-graph $D_{Q,NI}$. Consequently, the indexed set of cells adjacent to a given cell i (called the *neighbors* of i) are given by

$$N(i, NI) = \{h_1 i, h_2 i, \ldots, h_k i\} = NI\cdot i.$$

N(i,NI) is called the *neighborhood of cell* i. A (space state) *configuration* c: $Q \to A$ is an arbitrary assignment of states in A, to all the cells in the cellular space. The set of all configurations is called CON. The processor located at a given cell (also called a cell) is a finite state machine, and all processing elements are identical. If the RPN is in configuration c, then c(i) is the current state of cell i. The RPN operates in a synchronous manner, so c is interpreted as a specification of the state of the RPN at some time t. By the *state of the neighborhood* of cell i, we mean the indexed set c(N(i,NI)).

The operation of a RPN is specified by the application of a local function in set I to each cell of Q, producing the next state of the cell in terms of the state of its neighborhood. Thus, the state of any cell i at time t+1 is determined by the state of its neighbors at time t, and the particular local function in set I applied.

The invocation of the *same* local function $\sigma \varepsilon I$ to all cells of Q specifies a *parallel transformation* τ_σ, that maps the current configuration into the next. For each local function σ: $A^k \to A$, there is a unique parallel transformation τ_σ: CON $\to$ CON, and conversely. So for configurations c_1, c_2, $\tau_\sigma(c_1) = c_2$, $\sigma \varepsilon I$, if and only if for every cell i of Q, $c_2(i) = \sigma c_1 N(i,NI)$, where $\alpha\beta$ denotes $\alpha(\beta)$, by the above definitions. Configuration c_2 is called the *successor configuration* of c_1 and c_1 the *predecessor configuration* of c_2. The set PI = $\{\tau_\sigma | \sigma \varepsilon I\}$ is called the set of *admissible parallel transformations*. Since the same local function is applied to all cells, this mode of operation is called the *single-instruction-stream-multiple-data-stream* (SIMD) and the RPN said to be *uniform*. The terminology is understood by thinking of the set I of admissible local functions as a collection of hardware instructions built into each cell of the RPN; in the SIMD mode, the same "instruction" is executed at a given time on every cell in the network (see Flynn [51]).

If different local functions in set I are applied to different cells, the mode of processing is called *multiple-instruction-stream-multiple-data-stream* (MIMD).

When a regular processor network (A, Q, NI, I) is such that quasigroup Q is a group, it will be called a *strongly regular processor network* (REPN). If Q is the Abelian group constituted by the set of d-tuples of integers $\mathbb{Z}^d$, together with vector addition, the uniform RPN (A, Q, NI, I) corresponds to a "tessellation automaton" and will be referred to as a d-dimensional *tessellation processor network* (TPN).

Since a neighborhood index NI defines the neighbors of each cell of Q, NI may be considered a template that is moved around in the cellular space. The following neighborhood index of a one-dimensional TPN deserves a special name. A neighborhood index NI is said to be *scope*-k if NI = $\{j+1,\ldots,j+k\}$, for some integers j,k. Note that consecutive cells in this NI are adjacent. The scope indicates the number of cells included in the neighborhood.

Three neighborhood indices of a d-dimensional TPN have been referenced frequently in the literature. They are known as the *Moore*, *von Neumann* and *standard* neighborhood indices. For any $\varepsilon \geq 1$, a *Moore neighborhood index* J_ε is given by the set of d-tuples $\{\delta\}$ such that $||\delta|| \leq \varepsilon$, where $||\delta||$ is the maximum magnitude among the coordinates of d-tuple δ. For any $\varepsilon \geq 1$, a *von Neumann neighborhood index* H_ε is specified by the set of d-tuples $\{\delta\}$ such that $|\delta| \leq \varepsilon$, where $|\delta|$ is the sum of the magnitudes of the coordinates of d-tuple δ. Finally, a *standard neighborhood index* is obtained from H_ε by deleting all d-tuples δ in H_ε with some negative coordinate. For future reference and to illustrate the definitions, the first two of these interconnection structures of TPN's are reformulated in terms of their group-graphs. A group-graph $D_{G,NI}$, where $G = (NI,R)$, $K = \{a_1,\ldots,a_d\}$, $R = \{gh = hg, h \neq g\varepsilon K\}$ and $NI = \{e\} \cup \{h_1^{\nu_1} h_2^{\nu_2} \ldots h_b^{\nu_b} | b \leq d,\ 0 \leq \nu_n,\ n = 1,\ldots,b$, and mutually distinct $h_1,\ldots,h_b \in K \cup K^{-1}\}$, has a Moore topology J_ε if $0 \leq \nu_n \leq \varepsilon$, $n = 1,\ldots,b$, $b \leq d$; and a von Neumann topology H_ε if $0 \leq \nu_1 + \ldots + \nu_n \leq \varepsilon$. When no mention of subindex ε is made, it is understood that $\varepsilon = 1$.

Some applications require the inclusion in state alphabet A of two special states: the *quiescent state*, denoted by 0, and the *boundary state*, denoted by b. A cell in state 0 represents a processor that is temporarily idle. The state b is a tool for restricting the cellular space. Whenever state $0 \in A$, every local function σ in I is such that: $\sigma(0,0,\ldots,0) = 0$. Thus, a cell remains in the quiecent state as long as its neighbors are quiescent. A cell in state b remains permanently in that state, and since state b is in the range of no local function, it can only be assigned to a cell initially. An obvious use of state b is to specify a network with a finite number of processors, whenever the interconnection digraph of a RPN is infinite. Cells in the boundary state would then represent nonexistent processing elements. The set of cells not in state b constitutes the *retina* of a RPN. A configuration c is called *finite* if $c(i) \neq 0$ for a finite number of cells i, where 0 is the quiescent state. A local function σ such that $\sigma(0,0,..,0)=0$ is said to be *finite configuration preserving*, since if c is finite, then $\tau_\sigma(c)$ is also finite.

We say that two configurations c_1 and c_2 are *similar* if there exists an arc-color-preserving automorphism α of the underlying digraph $D_{Q,NI}$ of a RPN, such that for all cells i of $Q, c_1(\alpha(i)) = c_2(i)$. The equivalence classes of configurations determined by the relation of similarity are called *patterns*. Informally, any two state configurations that "look the same" when viewed from some pair $(i, \alpha(i))$ of cells in the RPN, constitute the same pattern. The concept of a pattern has a clear interest if RPN's are used for recognizing features in figures.

If the set of admissible local functions I of a RPN contains only one element, the RPN is called *monogenic*. If it contains at least two, the RPN is said to be *polygenic*. If $I = \{\sigma | \sigma\colon A^k \to A\}$, the RPN is termed *unrestricted*.

A (cellular) region Z is a subset of cellular space Q. A *partial configuration* c_Z is a mapping $c_Z\colon\ Z \to A$, for region $Z \subseteq Q$. Configuration $c\colon\ Q \to A$ is an *extension* of partial configuration $c_Z\colon\ Z \to A$, if $c_Z(p) = c(p)$ for all $p \in Z$; partial configuration c_Z is called a *restriction* of c. PCON denotes the set of all partial configurations. A partial configuration c_Z is *finite*, in case region Z is finite. Note that CON $\subseteq$ PCON.

The *successor region* SC(Z) of region $Z \subseteq P(D)$ in digraph D is the set $SC(Z) = \{v \in P(D) | \text{for all arcs } (u,v) \text{ of } D,\ u \in Z\}$. The *predecessor region* PR(Z) of $Z \subseteq P(D)$ is the set $PR(Z) = \{u \in P(D) | \text{for some arc } (u,v) \text{ of } D,\ v \in Z\}$.

A local function σ: $A^k \to A$, specifies a parallel transformation τ_σ: PCON $\to$ PCON in a uniform RPN, as follows. Given $c_{Z_1} : Z_1 \to A$, d_{Z_2}: $Z_2 \to A$ and σ: $A^k \to A$, then $\tau_\sigma(c_{Z_1}) = d_{Z_2}$, $\sigma \in I$, whenever $Z_2 = SC(Z_1)$; and there exist extensions c and d: $Q \to A$ of c_{Z_1} and d_{Z_2}, respectively, such that $\tau_\sigma(c) = d$, i.e., $d(i) = \sigma c N(i,NI)$ for all $i \in Q$.

A configuration $d \in$ CON is *Garden of Eden* (GOE) *with respect to parallel transformation* τ: CON $\to$ CON, in case no configuration $c \in$CON exists such that $\tau(c) = d$. A configuration is *Garden of Eden* (GOE) if it is GOE with respect to all admissible parallel transformations.

A partial configuration $d_{Z_2} \in$ PCON is GOE with respect to parallel transformation τ: PCON $\to$ PCON, whenever no configuration $c_{Z_1} \in$ PCON exists such that $\tau(c_{Z_1}) = d_{Z_2}$. A partial configuration is GOE, if it is GOE with respect to all parallel transformations in set $PI = \{\tau_\sigma | \sigma \in I\}$.

Let FCON and FPCON be the set of finite configurations and finite partial configurations, respectively. The *transition digraph* D_M of a uniform RPN M, consists of point set $P(D_M)$ = FCON and arc set $A(D_M) = \{(c,d) | d = \tau(c),\ \tau \in PI,\ c, d \in \text{FCON}\}$. The *general transition digraph* $\bar{D}_M$ of a uniform RPN M, consists of point set $P(\bar{D}_M)$ = FPCON and arc set $A(\bar{D}_M) = \{(c,d) | d = \tau(c),\ \tau \in PI,\ c, d \in \text{FPCON}\}$. A *primitive configuration* is any $c \in$ FCON such that $c(q) = a \in A$, for some $q \in Q$ and $c(p) = 0$, for all $q \neq p \in Q$.

A finite configuration c is GOE whenever point c of D_M is a *transmitter*, i.e., has indegree $id(c) = 0$. Similarly, finite partial configuration c_Z is GOE whenever point c_Z of $\bar{D}_M$ is of indegree $id(c_Z) = 0$.

A uniform unrestricted RPN M is *complete* in case every finite configuration of M is reachable in D_M from some primitive configuration. M is *transition-connected* if its transition digraph D_M is finitely connected. M is *general-transition-connected* if for all c_{Z_1}, c_{Z_2} in digraph $\bar{D}_M$, with $Z_2 = SC^{\theta}(Z_1)$, for some $\theta \geq 1$, c_{Z_2} is reachable from c_{Z_1}.

The dynamic behavior of a uniform RPN M may be analyzed by studying its associated digraphs D_M and $\bar{D}_M$. For example, define a *property* as a subset Π of $\dot{F}PCON$, and consider the subgraph D_Π of $\bar{D}_M$ induced by subset Π (i.e., the maximal subgraph with point set Π). A *sink* of a digraph is a point which is reachable from every other point. The set of all sinks S(D) of a digraph D induces a maximal connected subgraph of D, called a *component* of D, since the addition of any points or arcs of D to S(D) results in a subgraph which is not connected. Thus, no partial configuration outside set $S(D_\Pi)$ can be reached from one in $S(D_\Pi)$.

Let a property Π on $\bar{D}_M$ be such that $S(D_\Pi)$ is finite and nonempty. Suppose we desire to recognize if a given partial configuration c has property Π, by determining whether or not c can be transformed into a canonical form, through an appropriate sequence of parallel transformations. A parallel transformation τ is said to *preserve property* Π, in case $\tau(c) = d$ implies $c \in \Pi$ if and only if $d \in \Pi$. Partial configuration c will have property Π, whenever c may be reduced to a canonical form in set $S(D_\Pi)$, by some sequence of transformations that preserve Π. For instance, consider the class of partial configurations that represent a number "8." Assume the property-preserving transformation τ should perform a line-thinning process, and suppose the canonical forms s in $S(D_\Pi)$ are "two contiguous circles." If for every $c \in \Pi$, $s \in S(D_\Pi)$,

τ satisfies $d(\tau(c),s) < d(c,s)$, recognition is always achieved in a finite number of steps.

If a sequence of partial configurations Σ spells a walk in the general transition digraph $\bar{D}_M$ of a uniform RPN M, M is said to *generate sequence* Σ, *and* Σ to be *generable by* M. A set of sequences is generable by M, if each sequence in the set is generable by M. Generability is also called perfect output controllability or reproducibility in the field of control systems theory.

1.4. Objectives and Approaches

This dissertation attempts to provide a theoretical framework by means of which structural guidelines for large regularly interconnected processor networks are obtained. The main objective is to investigate the influence of the topology of a parallel processing system on its computational characteristics. A device called a RPN has been introduced for this purpose.

Regularity of the topology of a system is dictated both by considerations of practical significance and mathematical tractability. In view of this, our study focuses on networks composed of a set of identical processing elements connected together in a uniform manner.

Those topologies investigated are described by finitely connected quasiregular directed graphs. This notion is the most general one compatible with the assumed uniformity of connections. A finitely connected quasiregular directed graph is shown to be isomorphic to the quasigroup graph of some quasigroup. As a consequence, a generator set of the quasigroup constitutes a uniform neighborhood index for describing the interconnection structure of a RPN, a desirable property.

In Chapter II a number of dynamic questions concerning RPN's will be investigated. First, a necessary and sufficient condition for the existence of a Garden-of-Eden partial configuration in a RPN will be derived. Then, a method for the characterization of properties of a RPN at the functional level of its transition digraph will be introduced. The method is based on a propositional language, and involves testing solvability of Boolean equations. System theoretic questions of RPN's under various classes of constraints will be shown to be answerable with the procedure.

As an illustration of this method, a proof of the existence of a Garden-of-Eden finite configuration in a linear RPN shall be provided first. The general procedure for analyzing system theoretic questions will then be introduced. The characterization of system theoretic questions, for example, transition-connectedness and existence of Garden-of-Eden configurations, are of interest in that lower bounds on topologic system complexity can be generated from this characterization.

The concept of periodicity of a configuration will be shown to give an indication of the generating potential of the configuration. A lower bound on the time needed to produce an arbitrary configuration from a specified one will be proved, providing a measure for this potential.

A method for determining lower bounds on the complexity of the neighborhood structure and the number of hardware instructions necessary in a RPN that executes a given algorithm will be presented at the end of Chapter II. Different criteria of optimality will be discussed in relation to this problem.

In Chapter III, a definition of the topology of RPN's in terms of their neighborhood maps will be provided. This definition will then be compared with a formalization in terms of quasigroup-graphs.

Trade-offs in RPN's between hardware complexity and computational power, in relation to a representative class of computations, will be discussed next. This class is the set of problems executed within polynomial-time by nondeterministic Turing machines. It includes some problems for which it has been conjectured that no practical algorithms exist. These trade-offs are investigated in relation to three classes of RPN topology.

In many parallel processing systems, the possibility for decentralized interprocessor communication is important. In order to achieve this capability, information routing mechanisms must be devised. In Chapter IV, routing schemes that do not require knowledge of processor sites shall be analyzed.

First, the class of RPN's which admit schemes achieving monotonic convergence of a response message to the message sender shall be characterized. Next, the concepts of pseudomonotonic and strong message routing convergence will be introduced. Several conditions satisfied by digraphs that support these properties as well as some examples will be provided. Finally, some measures of routing performance will be given. These measures are useful in a comparative study of network topologies.

In Chapter V, we summarize our results and present a number of suggestions for future research.

CHAPTER II

ANALYSIS OF DYNAMIC PROPERTIES

Our intent in this chapter is to characterize the state transition digraphs of RPN's. Any satisfactory system must satisfy basic dynamic properties, for example, transition-connectedness. Thus, the analysis of a RPN at the functional level of its transition digraph will produce lower bounds on the complexity of an implementation.

First, a structural condition for the existence of a partial configuration that is not an image of any parallel transformation in a given uniform RPN will be determined. Then, a methodology for the characterization of dynamic properties in RPN's will be introduced. The methodology is based on a propositional language and involves testing the solvability of a Boolean equation. System theoretic properties of a RPN may be investigated with this methodology.

The concept of a periodic configuration is introduced. It shall be demonstrated that the degree of periodicity in a configuration relates to the ability of the RPN to generate other patterns when started from it. A lower bound on the time needed to generate an arbitrary configuration from a given one shall be derived, making this concept precise.

A method for determining lower bounds on the complexity of a uniform RPN that executes a given algorithm will be presented. Different criteria of optimality shall be discussed.

2.1. Existence of a Transmitter

A sufficient and necessary condition for the existence of a finite partial configuration that is Garden of Eden in a uniform monogenic RPN M (i.e., a transmitter in the general transition digraph $\bar{D}_M$ of M) is given in this section. A special case of this condition was established by Moore [25] and Myhill [26]. Their result applies to a two-dimensional TPN with a Moore topology J_1 (see Definition 1.7). Fundamental to the derivation of the condition are the new concepts of "tiling," "properness" and "admissibility." The following definitions (together with Definition 1.7) are in order.

DEFINITION 2.1

A *motion function* f: P(D) → P(D) on a digraph D is an automorphism of D. A *shift function* g on a quasigroup-graph $D_{Q,NI}$ is a motion function on $D_{Q,NI}$ which is arc-color preserving, i.e., for all $v \in Q$, $a \in NI$, g is bijective and g(av) = ag(v). A set of motion functions on D which constitutes a group under the operation of composition, is called a *motion group* on D. Two partial configurations c_{Z_1} and d_{Z_2} are called *similar or copies* (*modulo a motion group* F) whenever for some $f \in F$, $Z_2 = f(Z_1)$ and $d_{Z_2}(f(v)) = c_{Z_1}(v)$, $v \in Z_1$. Since F is a group, similarity is an equivalence relation. An equivalence class of similar partial configurations is termed a *partial pattern* (modulo F). A partial pattern containing c_Z is $[c_Z]$. Let D be a digraph, Z a subset of P(D) and F a motion group on D. (Z,F) is called a *tiling of* D if every point of D may be expressed uniquely as f(z), $f \in F$, $z \in Z$. If $f \neq f' \in F$, then $f(Z) \cap f'(Z) = \Phi$ (empty set); thus, (Z, F) partitions P(D).

The *successor region* SC(Z) of subset $Z \subseteq P(D)$ in digraph D is the set $SC(Z) = \{v \in P(D) | \forall(u,v) \in A(D),\ u \in Z\}$. The *predecessor region* $PR(Z)$ of $Z \subseteq P(D)$ is the set $PR(Z) = \{u \in P(D) | \exists(u,v) \in A(D),\ v \in Z\}$.

The following lemmas are necessary for proving our main theorem.

LEMMA 2.1

For an arbitrary digraph D and region $Z \subseteq P(D)$, (1) $SC(PR(Z)) \supseteq Z$; (2) $PR(SC(Z)) \subseteq Z$.

PROOF. $SC(PR(Z)) = \{v \in P(D) | \forall(u,v) \in A(D), \exists(u,z) \in A(D),\ z \in Z\}$. Thus, $v \in Z$ implies $v \in SC(PR(Z))$.

$PR(SC(Z)) = \{u \in P(D) | \exists(u,v) \in A(D),\ \forall(z,v) \in A(D),\ z \in Z\}$. Hence, $u \in PR(SC(Z))$ implies $u \in Z$. Δ

LEMMA 2.2

For a quasigroup-graph $D_{Q,NI}$ and region $Z \subseteq Q$,
(1) $SC(Z) = \{v \in Q | NI \cdot v \subseteq Z\}$; (2) $PR(Z) = \{NI \cdot z | z \in Z\}$.

PROOF. By definition of $D_{Q,NI}$, the neighbors of a given point v are given by $N(v,NI) = NI \cdot v$.

Thus, $SC(Z) = \{v \in Q | N(v,NI) \subseteq Z\} = \{v \in Q | NI \cdot v \subseteq Z\}$.

$PR(Z) = \{u \in Q | u \in N(z, NI),\ z \in Z\} = \{NI \cdot z | z \in Z\}$. Δ

We shall denote $NI \cdot Z = \{NI \cdot z | z \in Z\}$.

DEFINITION 2.2

A tiling (Z,F) with region Z and motion group F is said to be *strongly proper* with respect to parallel transformation τ: PCON $\rightarrow$ PCON, whenever if c_Z and $d_{f(Z)}$ are copies (modulo F), $f \in F$, then $\tau(c_Z)$ and $\tau(d_{f(Z)})$ are copies (modulo F), $c_Z, d_{f(Z)} \in$ PCON.

A tiling (Z,F) with region Z and motion group F is *proper* with respect to parallel transformation τ: PCON $\to$ PCON, in case $\tau(a_Z) = \tau(b_Z)$ whenever $\tau(c_{f(Z)}) = \tau(d_{f(Z)})$, for all $f \in F$, where $c_{f(Z)}$ is a copy of a_Z and $d_{f(Z)}$ a copy of b_Z (modulo F). A strongly proper tiling is proper.

Distinct partial configurations c_Z and $d_Z \in$ PCON are called (after Moore [25]) *mutually erasable* (ME) with respect to parallel transformation τ if

(1) $c_Z(i) = d_Z(i)$, $i \in$ BRD(Z), where SC(Q - (Z - BRD(Z))) = Q - SC(Z),

(2) $\tau(c_Z) = \tau(d_Z)$. Note that this is a relation of equivalence.

A digraph D is said to be *centered* in case SC(Z) $\subseteq$ Z, for all Z $\subseteq$ P(D).

A proper tiling is characterized by the following proposition.

PROPOSITION 2.1

A tiling (Z,F) of quasigroup-graph $D_{Q,NI}$ is proper with respect to parallel transformation τ_σ, in case $\sigma(x_{\gamma(1)},..,x_{\gamma(k)}) = \sigma(y_{\gamma(1)},..,y_{\gamma(k)})$ whenever $\sigma(x_1,..,x_k) = \sigma(y_1,..,y_k)$, $x_1,..,x_k$, $y_1,..,y_k \in A$, where σ is the local function associated with τ_σ, and γ: K $\to$ K = {1,..,k} is the permutation specified by

$$f((h_1,..,h_k)\cdot i) = (h_{\gamma(1)},..,h_{\gamma(k)})\cdot f(i),\ i \in SC(Z), f \in F,\ h_1,..,h_k \in NI.$$

PROOF. If (Z,F) is proper with respect to τ_σ, by definition of τ_σ, $\sigma c N(i,NI) = \sigma d N(i,NI)$ whenever $\sigma c N(f(j),NI) = \sigma d N(f(j),NI)$, $i \in SC(Z)$, $f(j) \in SC(f(Z))$. Since every $f \in F$ is an automorphism of $D_{Q,NI}$, $f(NI\cdot i) = (h_{\gamma(1)},..,h_{\gamma(k)})\cdot f(i)$, $i \in SC(Z)$, for some permutation γ:K $\to$ K = {1,..,k}. Hence, (Z,F) is proper in case $\sigma(x_{\gamma(1)},..,x_{\gamma(k)}) = \sigma(y_{\gamma(1)},..,y_{\gamma(k)})$ implies $\sigma(x_1,..,x_k) = \sigma(y_1,..,y_k)$, where $(x_1,..,x_k) = c(NI\cdot f(j))$, $(y_1,..,y_k) =$

$d(NI \cdot f(j)), f(j) \in SC(f(Z))$. Δ

The following Lemmas 2.3, 2.4 and 2.5 are essential to the proof of the main theorem.

LEMMA 2.3

For a centered quasigroup-graph $D_{Q,NI}$, $Z \subset Q$,

$$Z - X \subseteq BRD(Z), \text{ where } Q - Z = SC(Q - X).$$

PROOF. Since $Q - Z \subseteq Q - SC(Z)$, $SC(Q - X) \subseteq Q - SC(Z)$; hence, $SC(Q - X) \subseteq SC(Q - (Z - BRD(Z)))$. This implies $Q - X \subseteq Q - (Z - BRD(Z))$ and $Z - X \subseteq BRD(Z)$. Δ

DEFINITION 2.3

Let (Z,F) be a tiling of centered quasigroup-graph $D_{Q,NI}$ with infinite point set Q. (Z,F) is *admissible* if there exists an ascending sequence $F_1 \subset .. \subset F_\pi \subset .. \subset F$ satisfying

$$\lim_{\pi \to \infty} \frac{|Q_\pi - X|}{|Q_\pi|} = 0 \text{ for } \begin{array}{l} (1)\ X = SC(Q_\pi) \\ (2)\ X = INT(Q_\pi) \end{array}, \quad \text{where} \quad \begin{array}{l} SC(Q - INT(S)) = Q - S, \\ Q_\pi = \{f(Z) | f \in F_\pi\}. \end{array}$$

and $|S|$ is the cardinality of set S. Partial configuration c_{Z_1} is a *subconfiguration* of c_{Z_2}, if $Z_1 \subseteq Z_2$ and $c_{Z_1}(i) = c_{Z_2}(i)$, $i \in Z_1$.

LEMMA 2.4

Let (Z,F) be a tiling of centered quasigroup graph $D_{Q,NI}$, with respect to parallel transformation τ. If $c_{f(Z)} \text{ ME } d_{f(Z)}, \forall f \in F_\pi \subseteq F$, then $\tau(c_{Q_\pi}) = \tau(d_{Q_\pi})$ where $Q_\pi = \{f(Z) | f \in F_\pi\}$.

PROOF. By definition, $c_{f(Z)}$ ME $d_{f(Z)}$ whenever (1) $c_{f(Z)}(i) = d_{f(Z)}(i)$,

$i \in BRD(f(Z))$, and (2) $\tau(c_{f(Z)}) = \tau(d_{f(Z)})$, $f \in F_\pi$.

Case 1: Consider $\tau(c_{Q_\pi})(i)$, $i \in SC(f(Z))$. By Hypothesis 2, $\tau(c_{f(Z)})(i) = \tau(d_{f(Z)})(i)$, $f \in F_\pi$; thus, $\tau(c_{Q_\pi})(i) = \tau(d_{Q_\pi})(i)$, $i \in SC(f(Z))$.

Case 2: Suppose $i \in SC(Q_\pi) - SC(f(Z))$, for some $f \in F_\pi$, and let $j \in NI \cdot i$. If $j \in BRD(f(Z))$, $f \in F_\pi$, then by Hypothesis 1, $c_{f(Z)}(j) = d_{f(Z)}(j)$. Otherwise, if $j \notin f(Z)$, then $j \in f'(Z)$, $f' \neq f \in F_\pi$. Since $D_{Q,NI}$ is centered, $NI \cdot i \subseteq f'(Z) - X \subseteq BRD(f'(Z))$, where $Q - f'(Z) = SC(Q - X)$, by Lemma 2.3. Thus, $j \in BRD(f'(Z))$; consequently, again by Hypothesis 1, $c_{f'(Z)}(j) = d_{f'(Z)}(j)$. Hence, $c_{Q_\pi}(NI \cdot i) = d_{Q_\pi}(NI \cdot i)$, and $\tau(c_{Q_\pi})(i) = \tau(d_{Q_\pi})(i)$, $i \in SC(Q_\pi) - SC(f(Z))$, $f \in F_\pi$.

Cases 1 and 2 imply $\tau(c_{Q_\pi})(i) = \tau(d_{Q_\pi})(i)$, $i \in SC(Q_\pi)$; therefore, $\tau(c_{Q_\pi}) = \tau(d_{Q_\pi})$. Δ

LEMMA 2.5

Let (Z,F) be an admissible tiling of centered quasigroup-graph $D_{Q,NI}$ with infinite point set Q. If $\pi > \pi_1, \pi_2$, specified by $(a^z - 1)^{\pi_1} = a^{|SC(Q_{\pi_1})|}$ and $(a^z - 1)^{\pi_2} = a^{|SC(Q_{\pi_2})|}$, then $(a^z - 1)^\pi < a^{|SC(Q_\pi)|}$, $a^{|INT(Q_\pi)|}$, where $z = |Z|$, $\pi = |F_\pi|$, $Q_\pi = \{f(Z) | f \in F_\pi\}$, $F_1 \subset \ldots \subset F_\pi \subset \ldots \subset F$.

PROOF. Since (Z,F_π) partitions point set Q_π, $|Q_\pi| = |Z| \cdot \pi = z\pi$. Since (Z,F) is admissible, $\lim_{\pi\to\infty} |SC(Q_\pi)|/z\pi = \lim_{\pi\to\infty} |INT(Q_\pi)|/z\pi = 1$ (by Definition 2.3). Thus, $\lim_{\pi\to\infty} (a^z - 1)^\pi / a^{|SC(Q_\pi)|} = \lim_{\pi\to\infty} (a^z - 1)^\pi / a^{|INT(Q_\pi)|} = \lim_{\pi\to\infty} [(a^z - 1)/a^z]^\pi = 0$. Hence, $\pi > \pi_1, \pi_2$ implies $(a^z - 1)^\pi < a^{|SC(Q_\pi)|}$, $a^{|INT(Q_\pi)|}$. Δ

Note that Lemmas 2.1, 2.3, 2.4 and 2.5 are valid for a digraph such that the indegrees and outdegrees of its points are bounded, since such a digraph is obtained from a quasigroup-graph by removal of some arcs.

However, we are interested here in systems with uniform connection patterns only.

DEFINITION 2.4

An *ascending sequence of tilings* is an infinite collection of tilings (Z_1,F_1), (Z_2,F_2),..such that finite sets $Z_1,Z_2,\ldots$ satisfy $Z_1 \subset Z_2 \subset \ldots$.

THEOREM 2.6

Let $M = (A,Q,NI,I)$ be a uniform monogenic RPN with centered infinite digraph $D_{Q,NI}$. Suppose there exists an ascending sequence of admissible tilings of $D_{Q,NI}$ which are proper with respect to parallel transformation τ. The existence of two finite partial configurations that are mutually erasable with respect to τ, is a necessary as well as sufficient condition for the existence of a finite partial configuration that is a transmitter in the general transition digraph of M.

PROOF. Sufficiency: Consider an admissible tiling (Z,F) of $D_{Q,NI}$, that is proper with respect to parallel transformation τ: PCON $\rightarrow$ PCON. Suppose c_Z and $d_Z \in$ FPCON are mutually erasable with respect to τ, and let $Q_\pi = \{f(Z),\ f \in F_\pi \subset F\}$, $\pi > 1$.

Define the set $PCON_{Q_\pi} = \{x | x\colon Q_\pi \rightarrow A\}$, and consider the following equivalence relation RE on $PCON_{Q_\pi}$. Partial configurations c_{Q_π} RE d_{Q_π} whenever subconfigurations $c_{f(Z)}$, $d_{f(Z)}$ satisfy $c_{f(Z)}$ ME $d_{f(Z)}$ or $c_{f(Z)} = d_{f(Z)}$, $f \in F_\pi$, where $c_{f(Z)}(i) = c_{Q_\pi}(i)$, $d_{f(Z)}(i) = d_{Q_\pi}(i)$, $i \in f(Z)$. In other words, c_{Q_π} and d_{Q_π} are in the relation RE, in case each partial subconfiguration $c_{f(Z)}$ is either mutually erasable or identical to the corresponding partial subconfiguration $d_{f(Z)}$. Since tiling (Z,F) is proper with respect to τ, if c_Z ME d_Z, then $c_{f(Z)}$ ME $d_{f(Z)}$. Since c_Z ME d_Z, there are at most a^z-1 equivalence classes of mutually erasable configurations d: $Z \to A$, where $a = |A|$, $z = |Z|$. Thus, the number of equivalence classes of RE is at most $(a^z - 1)^{|F_\pi|}$, since $|F_\pi|$ is the number of regions f(Z) in Q_π, $f \in F_\pi$. If c_{Q_π} RE d_{Q_π} then $\tau(c_{Q_\pi}) = \tau(d_{Q_\pi})$, by Lemma 2.4. Hence, $|\tau(\mathrm{PCON}_{Q_\pi})| \leq (a^z - 1)^{|F_\pi|}$. The number of configurations in $\mathrm{PCON}_{SC(Q_\pi)}$ is $a^{|SC(Q_\pi)|}$, by definition of τ. Thus, if

$$(a^z-1)^{|F_\pi|} < a^{|SC(Q_\pi)|},$$

there exists y: $SC(Q_\pi) \to A$ such that $y \notin \tau(\mathrm{PCON}_{Q_\pi})$, since then $|\tau(\mathrm{PCON}_{Q_\pi})| < |\mathrm{PCON}_{SC(Q_\pi)}|$. Without loss of generality, let $|F_\pi| = \pi$. By Lemma 2.5, the inequality is satisfied when $\pi > \pi_1$, where $(a^z-1)^{\pi_1} = a^{|SC(Q_{\pi_1})|}$, $z = |Z|$. Since the size of $Z \subseteq Q$ is unbounded, the result follows.

Necessity: Assume there are no mutually erasable partial configurations. Then every two distinct partial configurations c_Y, d_Y such that $c_Y(i) = d_Y(i)$, $i \in Y-X = BRD(Y)$, where $SC(Q-X) = Q - SC(Y)$, have distinct successors. Thus, for any $c_Y, d_Y \in$ PCON, $\tau(c_Y) = \tau(d_Y)$ implies if $c_Y \neq d_Y$, then $c_{BRD(Y)} \neq d_{BRD(Y)}$. Hence, $|\{a_Y | \tau(a_Y) = b_{SC(Y)}\}| \leq a^{|BRD(Y)|}$. In other words, the number of partial configurations a_Y: $Y \to A$ which have a common image $b_{SC(Y)}$ under parallel transformation τ is no larger than $a^{|BRD(Y)|}$. Consequently, $|\tau(\mathrm{PCON}_Y)| \geq a^{|Y|}/a^{|BRD(Y)|} = a^{|Y| - |BRD(Y)|}$. Now let $SC(R) = Q_\pi$. Since $R - BRD(R) = INT(Q_\pi)$, where $SC(Q-INT(Q_\pi)) = Q-Q_\pi$, $|\tau(\mathrm{PCON}_R)| \geq a^{|R-BRD(R)|} = a^{|INT(Q_\pi)|}$.

For the sake of contradiction, suppose the existence of a finite partial configuration d_Z: $Z \to A$, where (Z,F) is a tiling, that is a transmitter in general transition digraph $\overline{D}_M$.

Every partial configuration y_Y, $Z \subseteq Y$, containing transmitter d_Z as a subconfiguration, is also a transmitter. To prove this, suppose there exists $x_X \varepsilon$ PCON such $\tau(x_X) = y_Y$. Then there exist extensions c_1, c_2:$Q \to A$ of x_X and y_Y, respectively, such that $\tau(c_1) = c_2$. Thus, there is a restriction c_C of c_1 such that $\tau(c_C) = d_Z$, since c_2 is an extension of d_Z, which contradicts the last assumption. Since d_Z is a transmitter, $|\tau(PCON_Z)| = a^z - 1$. Tiling (Z,F) is proper with respect to τ; thus, $|\tau(PCON_{f(Z)})| = a^z - 1$, for all $f \varepsilon F_\pi$, since each set $PCON_{f(Z)}$ contains a transmitter. Thus, $|\tau(PCON_R)| \leq (a^z - 1)^{|F_\pi|}$. Hence, $a^{|INT(Q_\pi)|} \leq (a^z - 1)^{|F_\pi|}$. However, for $\pi > \pi_2$, where $(a^z - 1)^{\pi_2} = a^{|INT(Q_{\pi_2})|}$, by Lemma 2.5 this inequality is contradicted. Thus, the inexistence of mutually erasable partial configurations implies the inexistence of transmitters. Δ

EXAMPLE 2.1

The "triangular" digraph $D_{Q,NI}$ with group presentation (NI,R), $NI = \{a_1, a_2, a_3\}$, $R = \{gh = hg,\ g \neq h \ \varepsilon\ NI,\ a_1a_2a_3 = e\}$, admits an ascending sequence of "rhombic" tilings (Z,F) with $Z = a_1^{\rho_1} a_2^{\rho_2}$, $-\eta \leq \rho_1, \rho_2 \leq \eta, \eta \geq 1$, $F(i) = \{a_1^{n\delta} a_2^{n\delta} i \mid \delta = 2\eta + 1$, for integers $n\}$, $i \varepsilon Q$. Tilings (Z,F) are proper, since F is a set of shift functions. Tilings (Z,F) are admissible, since rhombi $\{Q_\pi\} = \{a_1^{n\delta} a_2^{n\delta} Z \mid \delta = 2\eta + 1, -\pi \leq n \leq \pi\}$ satisfy $|Q_\pi| = (2\pi + 1)^2 \delta^2$, $|SC(Q_\pi)| = |INT(Q_\pi)| = ((2\pi + 1)\delta - 2)^2$; thus, $\lim_{\pi\to\infty}(|SC(Q\pi)|/|Q\pi|) = 1$.

Let parallel transformation τ_σ, where $A = \{0,1\}$ and σ: $A^3 \to A$ be specified by $\sigma(a_1,a_2,a_3) = 1$ except if $a_1 = a_2 = a_3 = 0$. Partial configuration b_Z specified by $b_Z(i) = 1$ at $e \neq i \in Z$, $b_Z(e) = 0$ is then a transmitter. Partial configurations c_Z, d_Z such that $c_Z(i) = 0$ at $e \neq i \in Z$, $c_Z(e) = 1$; $d_Z(i) = 1$ at all $i \in Z$, $\eta \geq 2$, are mutually erasable with respect to τ, since $\tau(c_Z) = \tau(d_Z)$ and $c_{BRD(Z)} = d_{BRD(Z)}$.

2.2. A Dynamic Characterization of RPN's

2.2.1. Preliminaries

In this section, a method for the examination of dynamic questions concerning RPN's is introduced. The procedure is based on a propositional language. This language expresses the transitional constraints of a particular RPN. The method involves testing the solvability of Boolean equations derived from the propositional language.

For simplicity, first we illustrate the concept of the method through a simple example. Consider whether a one-dimensional, scope-2, unrestricted, uniform TPN with $A = \{0,1\}$, $NI = \{(-1,0)\}$, has a finite configuration $c_2 \in$ FCON such that no $c_1 \in$ FCON and parallel transformation $\tau \in PI = \{\tau_\sigma | \sigma \in I\}$ exist which satisfy $\tau(c_1) = c_2$ (the terminology was given in Definition 1.7). Such finite configuration d is called Garden-of-Eden with universe FCON. Let c_1 and c_2 be two finite configurations, as follows:

$$c_1 = \bar{0}x_0x_1x_2\ldots x_m\bar{0}$$
$$c_2 = \bar{0}y_1y_2\ldots y_m\bar{0},$$

where $\bar{0}$ denotes a contiguous collection of cells in the one-dimensional cellular space and $x_i, y_i \in \{0,1\}$, $i=0,..,m$. The goal is to test whether there exists a configuration c_2, such that no c_1 can be found for which c_2 is a successor. We are interested in determining the smallest positive integer m characterizing the size of such a configuration c_2. Note that c_1 can be given the indicated form without loss of generality: if $c_1' = \bar{0}x_{-j}..x_0..x_m..x_{m+k}\bar{0}$, $j,k \geq 1$, has c_2 as a successor under the application of parallel transformation τ, then also c_1 has c_2 as a successor under the application of τ. Without loss of generality, we may set $y_1 = y_m = 1$ in configuration c_2. Every binary function σ: $\{0,1\}^2 \to \{0,1\}$ may be expressed as $\sigma(a,b) = \bar{a}bz_1 + a\bar{b}z_2 + abz_3$, where '+' denotes logical OR, '$\bar{\ }$' represents complementation, and $z_u, a, b \in \{0,1\}$, $u=1,2,3$. Hence, configuration $c_2 = y$ is a successor of configuration $c_1 = x$ under τ_σ whenever the following system of equalities is satisfied:

$$0 = x_0 z_1 \tag{0}$$
$$y_1 = \bar{x}_0 x_1 z_1 + x_0 \bar{x}_1 z_2 + x_0 x_1 z_3 \tag{2}$$
$$y_2 = \bar{x}_1 x_2 z_1 + x_1 \bar{x}_2 z_2 + x_1 x_2 z_3 \tag{4}$$
$$\vdots \qquad\qquad \vdots$$
$$y_{m-1} = \bar{x}_{m-2} x_{m-1} z_1 + x_{m-2} \bar{x}_{m-1} z_2 + x_{m-2} x_{m-1} z_3 \tag{5}$$
$$y_m = \bar{x}_{m-1} x_m z_1 + x_{m-1} \bar{x}_m z_2 + x_{m-1} x_m z_3 \tag{3}$$
$$0 = x_m z_2 \tag{1}$$
$$0 = z_1 \bar{z}_2 z_3 \tag{*}$$

The last equality (*) rules out the identity transformation. Note that the set of propositional variables PV used here is

$$PV = \{'x_i' \mid i \in Q\} \cup \{'y_j' \mid j \in Q\} \cup \{'z_u' \mid u=1,2,3\},$$

where the cellular space Q is the set of integers. Also note that the valuations of the implied language are: $x_i=1$ whenever $c_1(i)=x(i)=1$; $y_j=1$ whenever $c_2(j)=y(j)=1$; $z_u=1$ whenever local transformation z: $A^2 \to A$ satisfies $z(\urcorner u)=1$, where $i,j \in Q$, $u=1,2,3$ and $\urcorner u$ is the binary 2-tuple representing positive integer u.

The above system of equalities may be expressed as a single equivalent equality of the form $g(x,y,z) = 1$, where $x=(x_0,..,x_m)$, $y=(y_1,..,y_m)$, $z=(z_1,z_2,z_3)$; the expression $g(x,y,z)$ is called *representative* of the *transitional constraint*. To do this, recall that a system $r_1=s_1$, $r_2=s_2$,.. .,$r_\nu=s_\nu$,.. is equivalent to a single equality $(\bar{r}_1\bar{s}_1+r_1s_1)(\bar{r}_2\bar{s}_2+r_2s_2)..(\bar{r}_\nu\bar{s}_\nu+r_\nu s_\nu)..= 1$, for Boolean expressions r_ν, s_ν over alphabet PV.

Consider the Boolean equation $g(x,y,z) = 1$ such that y is a set of independent variables and x,z a set of unknowns. In other words, our goal is to express the x's and z's in terms of the y's. This equation, called the representative equation, is by construction unsolvable for some interpretation of y if and only if configuration $c_2=y$ is the successor of no configuration $c_1 = x$ (other than itself). For size $m \leq 3$ the set of such values of y is empty; for size m=4 such interpretations of y are $\bar{0}11010\bar{0}$ and $\bar{0}10110\bar{0}$. The method for deriving the set of values of the independent variables that make a Boolean equation unsolvable is explained in the next section.

By exploiting its symmetry, representative expression g may be derived by a recursion on its size m. The above system of equalities has been numbered according to the following indexing rule:

$$k = \begin{cases} 2j \text{ for } j=0,1,..,\lceil m/2 \rceil \\ 2(m-j+1) + 1 \text{ for } j=\lceil m/2 \rceil+1,..,m+1, \end{cases}$$

where j is the index in the left side of each equality. Let $w_k=1$, $k=0,\ldots,m+1$ denote equality number k; $w_*=1$ represent last equality (*); and $v^u = w_0\ldots w_k\ldots w_u$. Then, if $m=m_0$, representative expression g is given by $g = v^{m_0} w_{m_0+1} w_*$. Since $v^{m_0+1} = v^{m_0} w_{m_0+1}$, representative expression g for $m=m_0+1$ is obtained from representative expression g for $m=m_0$; thus, g may be derived by performing a recursion on its length m.

2.2.2. On Solving Boolean Equations

Given a Boolean equation of the form $f(x,y) = 0$ or $g(x,y) = 1$, where $x=(x_1,\ldots,x_r)$ is the set of unknowns and $y=(y_1,\ldots,y_s)$ the set of independent variables, we need to determine the set of values of y for which the equation is unsolvable. Two operators used for this are defined next.

DEFINITION 2.5

Let a function $h(x,y)$ be represented by an arbitrary set of implicants m_u (disjunctive normal form) or by an arbitrary set of implicates n_v (conjunctive normal form), where $\{x,y\}$ is a partition on the variables of h. Denote $m_u(x,y) = a_u(x)b_u(y)$, where $a_u(x)$ (respectively $b_u(y)$) is a logical product ("AND") of variables x (respectively y) or their complements; and $n_v(x,y) = c_v(x) + d_v(y)$, where $c_v(x)$ (respectively $d_v(x)$) is a logical sum ("OR") of variables x (respectively y) or their complements.

Operators $\forall$ and $\exists$ are defined by

$$(\forall x)h(x,y) = \prod_v d_v(y),$$

$$(\exists x)h(x,y) = \sum_u b_u(y),$$

for arbitrary function h so specified.

LEMMA 2.7 (Thayse [52])

Let $f(x,y) = 0$ or $g(x,y) = 1$, where $f = \bar{g}$, be a Boolean equation. The values of the variables y for which the equation is unsolvable are given by

$$(\forall x) f(x,y) = 1 \text{ or}$$
$$(\exists x) g(x,y) = 0.$$

EXAMPLE 2.2

We illustrate the deduction of the result presented in the last section. The terminology is the same.

The representative expression g is first derived for m=3.

$$v^3 = w_0 w_1 w_2 w_3 = \bar{x}_0 x_1 x_2 \bar{x}_3 z_1 z_2 + \bar{x}_0 x_1 x_2 x_3 z_1 \bar{z}_2 z_3 + x_0 x_1 x_2 \bar{x}_3 \bar{z}_1 z_2 z_3 + x_0 x_1 x_2 x_3 \bar{z}_1 \bar{z}_2 z_3;$$

Since

$$g = v^3 w_4 w_*,$$

$$g(x,y,z) = (\bar{x}_0 x_1 x_2 \bar{x}_3 z_1 z_2 z_3 + x_0 x_1 x_2 \bar{x}_3 \bar{z}_1 z_2 z_3 + x_0 x_1 x_2 x_3 \bar{z}_1 \bar{z}_2 z_3)(y_2 + \bar{y}_2);$$

So $(\exists x,z) g(x,y,z) = y_2 + \bar{y}_2 = 1$; hence, the representative equation, if m=3, is solvable for every value of the variables y.

Setting m=4, we obtain $(\exists x,z) g(x,y,z) = \bar{y}_2 \bar{y}_3 + y_2 y_3$. This proves that patterns $\bar{0}1101\bar{0}$ and $\bar{0}1011\bar{0}$ have no predecessor finite configuration in the given RPN.

2.2.3. The General Method

The approach suggested by the example in Section 2.2.1 is now formalized. Although it will be discussed in relation with the dynamic questions to be given in Def. 2.2 only, it may characterize any properties which admit a description in terms of a propositional language.

DEFINITION 2.6

A propositional language for analyzing dynamic questions of uniform RPN's is introduced as follows. The syntax of the language is determined by the set of propositional variables PV, defined by:

$$PV = \{'x_{i,\nu}' \mid i\varepsilon Q,\ \nu=1,\ldots,\alpha\} \cup \{'y_{j,\nu}' \mid j\varepsilon Q,\ \nu=1,\ldots,\alpha\} \cup$$

$$\{'z^{t}_{u,\nu}' \mid u\varepsilon U,\ \nu=1,..,\alpha,\ t \geq 1\}$$

where $\alpha = \lceil \log_2 a \rceil$; $a=|A|$ is the cardinality of state alphabet A; $k = |NI|$;

$$U = \{\text{integer } u \geq 0 \mid (k\alpha)\rceil u = \rceil s_1 \ldots \rceil s_k,\ 0 \leq s_1,..,s_k \leq a-1\};$$

$\lceil v \rceil$ is the smallest integer $\geq v$; $(k\alpha)\rceil u$ is the binary $k\alpha$-tuple representing nonnegative integer u; and $\rceil u = (\alpha)\rceil u$.

The semantics of the language is given by the following set of valuations on PV:

$x_{i,1}..x_{i,\nu}..x_{i,\alpha} = \rceil s$ whenever in partial configuration $x_{X(\mu,\theta)}$, $x_{X(\mu,\theta)}(i) = s$, $s=0,..,a-1$, $i\varepsilon X(\mu,\theta)$, $SC^{\mu+\theta}(X(\mu,\theta)) =$ NI, $\mu \geq 0$, $\theta \geq 1$;

$y_{i,1}..y_{i,\nu}..y_{i,\alpha} = \rceil s$ whenever in partial configuration $y_{Y(\mu)}$, $y_{Y(\mu)}(j) = s$, $j\varepsilon Y(\mu)$, $s=0,..,a-1$, $SC^{\mu}(Y(\mu)) =$ NI, $\mu \geq 0$, $\theta \geq 1$;

$z^{\theta}_{u,1}..z^{\theta}_{u,\nu}..z^{\theta}_{u,\alpha} = \rceil s$ whenever admissible local function z^{θ}: $A^k \rightarrow A$ in I satisfies $z^{\theta}(s_1,..,s_k) = s$, where

$$(k\alpha)\top u = \top s_1 \ldots \top s_k, \quad 0 \le s_1, \ldots, s_k \le a-1, \quad \theta \ge 1.$$

Partial configuration $x_{X(\mu,\theta)}: X(\mu,\theta) \to A$ represents the initial point, and partial configuration $y_{Y(\mu)}: Y(\mu) \to A$ the final point in a sequence $z^\theta, z^{\theta-1}, \ldots, z^1$ of admissible local functions producing $y_{Y(\mu)}$ from $x_{X(\mu,\theta)}$.

This statement, expressed formally by $\tau_{z^1}\tau_{z^2}\ldots\tau_{z^\theta}(x_{X(\mu,\theta)}) = y_{Y(\mu,\theta)}$ is equivalent to the following system of equalities:

$$y_{j,\nu} = \sum_{u\in U} m^\theta_{u,j}(x) z^\theta_{u,\nu}, \text{ for } j\in Y(\mu,\theta),\ \nu=1,2,\ldots,\alpha,$$

and U as above, where $m^\theta_{u,j}(x)$ is recursively defined by

$$(1)\quad m^1_{u,j}(x) = x^{u_1}_{h_1\cdot j,1} \ldots x^{u_{k\alpha}}_{h_k\cdot j,\alpha}, \quad NI = \{h_1, h_2, \ldots, h_k\},$$

for binary exponents $u_1 \ldots u_{k\alpha} = (k\alpha)\top u$, $\omega^{u_1} = \bar{\omega}$ if $u_1 = 0$, $\omega^{u_1} = \omega$ otherwise, and Boolean expression ω.

$$(2)\quad m^{\theta+1}_{u,j}(x) = \xi^{u_1}_{h_1\cdot j,1} \ldots \xi^{u_{k\alpha}}_{h_k\cdot j,\alpha},$$

where

$$\xi_{j,\nu} = \sum_{u\in U} m^\theta_{u,j}(x) z^\theta_{u,\nu}.$$

Expression $g_{\mu,\theta}(x,y,z)$ such that $g_{\mu,\theta}(x,y,z)=1$,

for $\quad x = \{x_{i,\nu} | i\in X(\mu,\theta), 1 \le \nu \le \alpha\}$, $y = \{y_{j,\nu} | j\in Y(\mu), 1 \le \nu \le \alpha\}$,

$z = \{z^t_{u,\nu} | 1 \le t \le \theta, u \in U, 1 \le \nu \le \alpha\}$ is equivalent to the above system of equalities, is called the *representative expression* of the RPN.

The following lemma provides a recursive characterization of $g_{\mu,\theta}$.

LEMMA 2.8

The representative expression $g_{\mu,\theta}$ of a RPN is defined recursively as follows.

(1) $g_{0,1}$: $y_{j,\nu} = \sum_{u \varepsilon U} m^1_{u,j}(x) z^1_{u,\nu}$ for $j \varepsilon NI$;

(2) $g_{\mu+1,1} = g_{\mu,1} w_{\mu+1}$,

where $w_{\mu+1}$: $y_{j,\nu} = \sum_{u \varepsilon U} m^1_{u,j}(x) z^1_{u,\nu}$ for $j \varepsilon Y(\mu+1,1) - Y(\mu,1)$;

(3) $g_{0,\theta+1}(x,y,z) = g_{0,\theta}(x',y,z)$,

where $x'_{j,\nu} = \sum_{u \varepsilon U} m^\theta_{u,j}(x) z^\theta_{u,\nu}$, $j \varepsilon Y(0,\theta)$, $\nu = 1,2,..\alpha$, $\mu \geq 0$, $\theta \geq 1$.

PROOF. It follows from Definition 2.6. Δ

DEFINITION 2.7

A finite partial configuration $c_C \; \varepsilon$ FPCON is *Garden-of-Eden* whenever it is a transmitter in general transition digraph $\bar{D}_M$ of uniform RPN M.

A uniform RPN M is *θ-step general-transition-connected*, $\theta \geq 1$, if for all c_{C_1}, c_{C_2}, with $C_2 = SC^\theta(C_1)$, c_{C_2} is reachable from c_{C_1}. M is *general transition-connected* if it is θ-step general-transition-connected for some finite positive integer θ.

A function $\alpha(n)$ is said to be of *order* $\kappa(n)$ (written $O(\kappa(n))$) if there exists a constant δ such that $\alpha(n) \leq \delta\kappa(n)$ for all but some finite set of nonnegative integer values for n.

LEMMA 2.9

An upper bound on the asymptotic worst-case time complexity of an algorithm for evaluating a disjunctive normal form of representative expression $g_{\mu,\theta}$ is δ^n, where $n = \alpha(|X(\mu,\theta)| + |Y(\mu)| + \theta a^k)$, $\delta > 1$, $\alpha = \lceil \log_2 a \rceil$, $a = |A|$ and $k = |NI|$.

PROOF. The number of variables in $g_{\mu,\theta}(x,y,z)$ is $\alpha|X(\mu,\theta)| + \alpha|Y(\mu)| + \alpha\theta\ a^k$, where the first term constitutes the number of x's, the second term the number of y's and the third term the number of z's. The problem of determining the satisfiability of an arbitrary Boolean expression of n variables has been proved to be NP-complete (see Aho, Hopcroft and Ullman [62]). Thus, probably no algorithm which derives a disjunctive normal form for $g_{\mu,\theta}$ uses worst-case time equal to a polynomial of $n' = \alpha(|X(\mu,\theta)| + |Y(\mu)|)$. An upper bound for evaluating $g_{\mu,\theta}$ is $O(\delta^n)$ (see [53]), for some constant δ. Δ

THEOREM 2.10

A uniform unrestricted RPN has no finite partial configuration $c_C \in$ FPCON that is Garden-of-Eden, where $SC^\mu(C) \subseteq NI$, if and only if $(\exists x,z)g_{\mu,1} = 1$ for $\mu \geq 1$.

PROOF. By Definition 2.6, finite partial configuration $y_{Y(\mu)}$ is the successor of $x_{X(\mu,1)}$ by application of local function z, whenever $g_{\mu,1}(x,y,z) = 1$, $SC^\mu(Y(\mu)) = NI$, $SC^{\mu+1}(X(\mu,1)) = NI$. Thus, in case there exists a value of y such that there are no values of x and z satisfying $g_{\mu,1}(x,y,z) = 1$, no predecessor for $y_{Y(\mu)}$ will exist. By Lemma 2.7, this occurs if and only if $(\exists x,z)g_{\mu,1} \neq 1$. Δ

A partial algorithm for determining whether a RPN has a Garden-of-Eden finite partial configuration is suggested by Theorem 2.10. Namely, find the smallest μ such that $(\exists x,z)g_{\mu,1} \neq 1$. The recursive nature of $g_{\mu,1}$, illustrated in Lemma 2.8, allows the computation of $(\exists x,z)g_{\mu+1,1}$, based on the computation of the expression for $g_{\mu,1}$. Parsing algorithms developed for the efficient evaluation of disjunctive normal forms of Boolean expressions may be applied to reduce the time needed to compute $(\exists x,z)g_{\mu,1}$ for a given RPN. A discussion on the expected time complexity of these algorithms is given by Aho and Ullman [53] and Karp [54].

THEOREM 2.11

Let M be a uniform unrestricted RPN. For all finite partial configurations c_{C_1}, c_{C_2} in M, with $C_2 = SC^{\theta}(C_1)$, and $SC^{\mu}(C_1) \subseteq NI$, c_{C_2} is reachable from c_{C_1} if and only if $(\exists z)g_{\mu,\theta} = 1$.

PROOF. By Definition 2.6, finite partial configuration $y_{Y(\mu)}$ is the successor of $x_{X(\mu,\theta)}$ by application of a sequence of local functions of z^{θ}, $z^{\theta-1}, \ldots, z^{1}$, whenever $g_{\mu,\theta}(x,y,z) = 1, SC^{\mu}(Y(\mu)) = NI$, $SC^{\mu+\theta}(X(\mu,\theta)) = NI$. Thus, in case there exist values of x,y such that there are values of z satisfying $g_{\mu,\theta}(x,y,z) = 1$, $y_{Y(\mu)}$ will be reachable from $x_{X(\mu,\theta)}$. By Lemma 2.7, this happens in case $(\exists z)g_{\mu,\theta} = 1$. Δ

A partial algorithm for determining whether a RPN is general transition-connected is suggested by this theorem. That is, find the smallest μ,θ such that $(\exists z)g_{\mu,\theta} = 1$. By Lemma 2.8, the computation of $(\exists z)g_{\mu+1,\theta+1}$ can make use of the expression for $g_{\mu,\theta}$. This computation may be speeded up for particular RPN's by exploiting their specific structure, as discussed above. In some cases it might be sufficient to determine whether a RPN is θ-step general-transition-connected for $\theta(NI,A)$, for all finite partial configurations of bounded size given by $\theta(NI,A)$. The minimum positive integer θ such that RPN M is general-transition-connected provides a worst-case lower time bound on the operation of M.

The set of partial configurations may, in addition, be stipulated to preserve a given set of properties Π. The properties are incorporated to the transitional system of equalities by adding equalities of the form $h_\gamma(x,z) = 1$, $\gamma \in \Pi$ to it. Properties of interest, especially to applications in pattern processing, include connectedness, convexity, fixed Euler number, monotonic convergence to a set of goal configurations (see Smith [37]).

DEFINITION 2.8

A *sink* of a digraph is a point which is reachable from every other point. The set of all sinks S(D) of a digraph D induces a maximal connected subgraph of D, called a *component* of D, since the addition of any points or arcs of D to S(D) results in a subgraph which is no longer connected. Let a *property* Π be a subset Π of FPCON. Consider the subgraph D_Π of $\bar{D}_M$ induced by subset Π (i.e., the maximal subgraph with point set Π). No partial configuration outside set $S(D_\Pi)$ is reachable from a partial configuration in $S(D_\Pi)$. On page 37 of Chapter I, an application was mentioned in which determining the set of all sinks is of interest.

By definition, there are no cycles in general transition-digraph

$\bar{D}_M$. Such a digraph is called *acyclic*. A *source* of a digraph is a point from which every other point is reachable.

System theoretic properties of a RPN M may be characterized by the formulation of graph theoretic properties of its general transition digraph $\bar{D}_M$, obtained from the representative expression $g_{\mu,\theta}$ of M.

Thus, evaluating the shortest path between a given pair of partial configurations (whenever it exists); establishing whether a particular partial configuration is a source (or a sink); finding out whether a given sequence of partial configurations is generable, are all properties of a subdigraph D_Π of D that may be derived in a manner analogous to the one applied in Theorems 2.10 and 2.11.

Giving timing constraints as a function of network topology, state set cardinality and instruction set size, provides a designer with comparative guidelines that might be of help in selecting a system organization that will meet his computational requirements.

A refinement in the specification of a RPN that provides a better characterization of a system at additional computational expense, is to partition the set of admissible local maps I (i.e., the set of instructions) into two subsets, one performing state-transition functions and the other effecting information routing functions. This would allow distinguishing the data switching requirements from the strictly computational provisions in the output produced by the algorithms of this section.

2.3. Periodic Configurations in Uniform RPN's

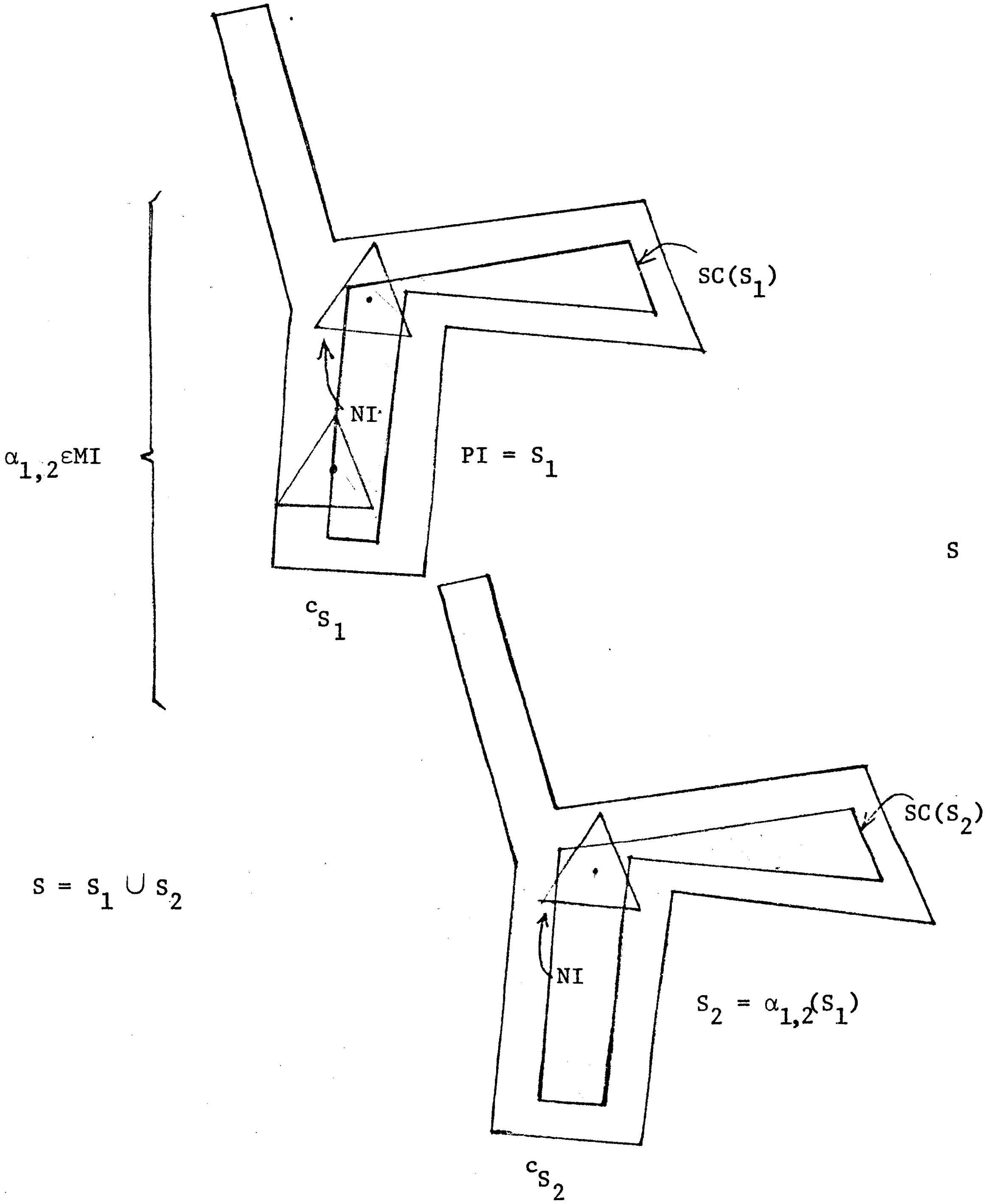
In this section, the concept of a periodic configuration is defined. For convenience, some definitions of Section 2.1 are duplicated.

DEFINITION 2.9

A *motion function* f: P(D) → P(D) on a digraph D is an automorphism of D. A *shift function* g on a quasigroup-graph $D_{Q,NI}$ is a motion function on $D_{Q,NI}$ which is arc-color preserving, i.e., for all $u,v \in Q$, $a \in NI$, g is bijective and $g(av) = ag(v)$. A set of motion functions on D which constitutes a group under the operation of composition, is called a *motion group* on D. Two partial configurations c_{Z_1} and d_{Z_2} are called *similar or copies (modulo a motion group* F*)* whenever for some $f \in F$, $Z_2 = f(Z_1)$ and $d_{Z_2}(f(v)) = c_{Z_1}(v)$, $v \in Z_1$. Since F is a group, similarity is an equivalence relation. An equivalence class of similar partial configurations is termed a *partial pattern* (modulo F). A partial pattern containing partial configuration c_Z is denoted $[c_Z]$.

A *periodic fragment* c_S of a configuration c is a partial configuration c_S: S → A, for $S \subseteq Q$, such that S is the maximal union of distinct subsets S_1, S_2,... satisfying: for every two subsets S_π, S_ρ, $\pi \neq \rho$, there exists a motion function α such that if c_S and $d_{\alpha(S)}$ are copies (modulo α), then $\tau(c_S)$ and $\tau(d_{\alpha(S)})$ are copies (modulo α), $\tau \in PI$, $c_S, d_{\alpha(S)} \in PCON$. If the number of subsets S_1, S_2,.. composing the set S is minimum, subset S_1 is called a *periodic index* PI. A periodic index PI is a template that repeats itself, and set S indicates the region of periodicity. A configuration may have several periodic fragments. The set of motion functions α satisfying the above condition is called the *motion index* MI of periodic region S; S is denoted (PI,MI).

The above definitions are illustrated in Figure 2.1.



PERIODIC FRAGMENT (PI, $\{\alpha_{1,2}\}$)

Figure 2.1. A Periodic Configuration

The following lemma provides limits that characterize the evolution of a periodic fragment throughout time.

LEMMA 2.12

Let M be a uniform RPN with neighborhood index NI. Let c_1 be a configuration with periodic fragment c_{S_1} such that $S_1 = (PI_1, MI_1)$, and c_2 any successor of c_1 with periodic fragment c_{S_2}, where $S_2 = (PI_2, MI_2)$. Then $MI_2 \supseteq MI_1$, and $S_2 \supseteq SC(S_1)$.

PROOF. Let $W = \{i | NI \cdot i \subseteq S_1\} = SC(S_1)$. If $i \in W$, then by Definition 2.9, $c_1(NI \cdot i) = c_1(\alpha(NI \cdot i))$, for some $\alpha \in MI_1$ and thus $c_1(N(i,NI)) = c_1(\alpha(N(i,NI)))$. So by definition of $\alpha, c_2(i) = c_2(\alpha(i))$, and thus $i \in S_2$. Hence, $\alpha \in MI_2$ and $W \subseteq S_2$. Δ

The following lemma illustrates the role of the neighborhood index in the temporal evolution of a periodic fragment.

DEFINITION 2.10

A partial configuration c_S in a uniform RPN is *periodic-constraint-free* if $c_S(N(i_1,NI)) = c_S(N(i_2,NI))$ implies $i_1 = i_2, i_1, i_2 \in SC(S)$. Informally, a partial configuration S is periodic-contraint-free whenever every two neighborhoods in SC(S) are mutually distinct. Partial configurations with periodic constraint have a more restricted set of successor configurations than periodic-constraint-free configurations.

LEMMA 2.13

Let M be a uniform RPN M with neighborhood index NI.
Let c be a configuration in M with periodic fragment c_S, defined by

$S = (PI, MI)$. If $SC(S) = \phi$, then $c_S(N(i,NI)) \neq c_S(N(\alpha(i),NI))$ for every $i \in SC(S)$, $\alpha \in MI$.

PROOF. Assume $N(i,NI) \not\subseteq S$ for all $i \in Q$. Let $c(N(i,NI)) = c(N(\alpha(i),NI))$ for some $i \in Q$ and $\alpha \in MI$. Then, set $S \cup N(i,NI)$ defines a larger periodic fragment of c with motion index MI. This is a contradiction, since by Definition 2.9, set S is maximal. As a consequence, $c(N(i,NI)) \neq c(N(\alpha(i),NI))$ for all $i \in SC(S)$. Δ

A lower bound on the time required to reach an arbitrary configuration from a given periodic one is provided by the following theorem.

THEOREM 2.14

Let M be a uniform RPN with quasigroup graph $D_{Q,NI}$. Let c be a configuration in M with periodic fragment c_S, where $S = (PI, MI)$. Assume θ is the largest positive integer such that $N^{\theta}(i) \subseteq S$, for some cell $i \in Q$, where (1) $N^0(i) = i$ and (2) $N^{\theta+1}(i) = NI\cdot(N^{\theta}(i))$, $S\cdot T = \{s\cdot t | s \in S, t \in T\}$. Then, there exists a configuration $\bar{c}$ which is not reachable from c within θ transitions.

PROOF. Define $W_1 = SC(S)$ and $W_{\theta+1} = SC(W_{\theta})$, $\theta \geq 1$. Let configuration c_{θ} with periodic fragment $c_{S_{\theta}}$ be a successor of configuration $c_{\theta-1}$ with periodic fragment $c_{S_{\theta-1}}$, $\theta \geq 1$, $c_0 = c$, $S_0 = S$. By Lemma 2.12 $S_{\theta} \supseteq W_{\theta}$, $\theta > 1$.

Next, $W_{\theta} = SC^{\theta}(S) = \{i \in Q | N^{\theta}(i) \subseteq S\}$. To show this, an induction on θ is effected. By definition, $W_1 = \{i \in Q | N^1(i) \subseteq S\}$. Let it be true for $1 \leq t \leq \theta$. Then, the following statements are equivalent: $i \in W_{\theta+1}$; $h\cdot i \in W_{\theta}$; $N^{\theta}(h\cdot i) \subseteq S$, for $h \in NI$; $N^{\theta+1}(i) \subseteq S$. Thus,

$W_{\theta+1} = \{i \in Q | N^{\theta+1}(i) \subseteq S\}$. Hence, $N^{\theta}(i_{\theta}) \subseteq S$ implies that $c_t(i_t) = c_t(\alpha(i_t))$, for $\alpha \in MI$, $i_t \in W_t$, $1 \leq t \leq \theta$, $i_0 = i$. Therefore, configuration $\bar{c}$ such that $\bar{c}(i_t) \neq \bar{c}(\alpha(i_t))$, $N^t(i_t) \subseteq S$, $1 \leq t \leq \theta$, is not reachable from c within θ transitions. Δ

The theorem indicates the larger the periodic fragment of a configuration is, the longer it takes to generate some arbitrary configuration from it. If the size $|Z|$ of a partial configuration $c_Z: Z \to A$ increases sufficiently, partial configuration c_Z can not be periodic-constraint-free. The following theorem makes this precise.

THEOREM 2.15

Let $M = (A,Q,NI,I)$ be a uniform RPN. A largest periodic-constraint-free partial configuration c_Z in M is of size $|Z| = ka^k$, where $k = |NI|$, $a = |A|$.

PROOF. Consider a cellular region Z specified as follows: $Z = \{NI \cdot i | NI \cdot i_{\pi} \cap NI \cdot i_{\rho} \neq \phi \Rightarrow i_{\pi} = i_{\rho},\ i,\ i_{\pi}, i_{\rho} \in Q\}$. Let c_Z be a partial configuration $c_Z: Z \to A$ that satisfies $c_Z(NI \cdot i_{\pi}) = c_Z(NI \cdot i_{\rho}) \Rightarrow i_{\pi} = i_{\rho}$. Partial configuration c_Z is a union of mutually disjoint and distinct neighborhoods. Thus, c_Z is periodic-constraint-free.

If $|SC(Z)| = a^k$, then $|Z| = ka^k$. There are a^k distinct maps from NI onto A. Thus, any partial configuration c_Y such that $|Y| > |Z|$ will contain at least two identical neighborhoods; therefore such a c_Y is not periodic-constraint-free. Δ

DEFINITION 2.11

A partial configuration $c_Z: Z \to A$ is called *privileged* if

$|\{c_Z(N(i,NI))|i \in Z\}| = a^k$. A set of configurations C is *finitely arbitrary* in case for all finite $r \geq 1$, $i_1,\ldots,i_r \in Q$ and $s_1,\ldots,s_r \in A$, there is a configuration c in C that satisfies $c(i_j) = s_j$ for $1 \leq j \leq r$. Privileged configurations are finitely arbitrary.

Grosky [55] has established that if there are no finitely arbitrary Garden-of-Eden configurations, there are none at all. Thus, in order to determine whether there exists a configuration with no predecessors, it suffices to examine the set of privileged configurations only. Therefore, the algorithm presented in previous section for determining Garden-of-Eden configurations may be speeded up by imposing the constraint of privilege on the representative equation $g_{\mu,1}$.

A lower bound for the size of a privileged partial configuration c_Z is a^k, since $|SC(Z)| \geq a^k$.

Theorem 2.14 has illustrated that configurations with a lower periodic constraint have a relatively higher generating potential, in the sense that they are predecessors of a relatively larger set of patterns. Also, configurations with a low periodic constraint are generated by a relatively smaller set of patterns.

2.4 Structural Conditions for Generability in RPN's

Given the specification Σ of a task, to determine whether some system M is suitable for executing this task is a problem we shall call that of generability of the specification Σ by the system M. In the context of this dissertation, the problem may be formulated as that of finding out whether a given set Σ of state configurations sequences can be produced by some appropriate RPN M. This latter problem is trivial if the RPN M operates in the MIMD mode, i.e., if an arbitrary local function may be used at a given time step in each processor of M. In view of this, the investigation of the question shall be restricted to uniform RPN's. A condition for the generability of some set of sequences Σ by some RPN M will be given in Theorem 2.16.

Given a set Σ of state configuration sequences, it is of value to investigate the necessary structural conditions of a uniform RPN M which produces these sequences Σ. This question will be approached by providing conditions for optimality of M under various criteria of structural cost. A key concept in this characterization will be that of a prime neighborhood index.

DEFINITION 2.12

Given $S = c_1, c_2, .., c_t, ..$, a sequence of configurations in set CON, let ζ: CON $\rightarrow$ CON be the correspondence specified by $\zeta(c_t) = c_{t+1}$, $t \geq 1$. Correspondence ζ is said to be *associated* with sequence S. Let $P \subseteq Q$ be a finite region, and z^t: $A^{|P|} \rightarrow A$ be the correspondence specified by $z^t(c_t(P \cdot i)) = c_{t+1}(i)$, for all $i \in Q$, $t \geq 1$. Correspondence z^t is said to be *originated* at step t by set P and sequence S.

A RPN with neighborhood index NI produces sequence S only if each correspondence z^t originated at every step t by set NI and sequence S, $t \geq 1$, constitutes a function. Such a set NI is called *acceptable* with respect to sequence S. A set NI is acceptable with respect to a set of configuration sequences Σ if NI is acceptable with respect to each sequence $S \in \Sigma$. More formally, NI is acceptable with respect to sequence S if $c_t(NI \cdot i) = c_t(NI \cdot j)$ implies $c_{t+1}(i) = c_{t+1}$, for all $i, j \in Q$.

Set $NI \subseteq Q$ is said to be *prime* with respect to set of sequences Σ if (1) NI is acceptable with respect to Σ and (2) no set $P \subseteq NI$ satisfies (1).

Theorem 2.16 will provide an analytic condition for the generability of a set of sequences Σ by a RPN with neighborhood index NI, such that NI is prime with respect to Σ. First, the following terminology is introduced.

DEFINITION 2.13

A correspondence $z: A^k \to A$ is represented by the set of binary variables $\{'z_s' \mid s \varepsilon A\}$, such that $z_s = 1$, $z_r = 0$, $r \neq s$, whenever correspondence z satisfies $z(s_1, \ldots, s_k) = s$, $s_1, \ldots, s_k, r, s \varepsilon A$. The contraction $(\exists s_u)f: A^{k-1} \to \{0,1\}$ of correspondence $f: A^k \to \{0,1\}$ is specified by $(\exists s_u)f(s_1, .., s_{u-1}, s_{u+1}, .., s_k) = \sum_{s_u \varepsilon A} f(s_1, .., s_{u-1}, s_u, s_{u+1}, .., s_k)$. Note the equivalence of the contraction operator with the operator introduced in Definition 2.5, if f is a binary function.

THEOREM 2.16

A set of sequences Σ is generated by a uniform RPN with neighborhood index NI, and NI is prime with respect to Σ, if and only if conditions

(1) $\sum_{r \neq s \varepsilon A} z_r^t z_s^t = 0$ and

(2) $\sum_{r \neq s \varepsilon A} (\exists s_u) z_r^t(s_1, .., s_u, .., s_k) \cdot (\exists s_u) z_s^t(s_1, .., s_u, .., s_k) \neq 0$

are satisfied by correspondence z^t originated at step t by NI and Σ, $t \geq 1$, for all $s_u \varepsilon A$, $k = |NI|$.

PROOF. By definition of the binary representation of z^t, condition (1) indicates whether correspondence z^t is a mapping. Condition (2) is verified whenever no component $h_u \varepsilon$ NI, $u = 1,2,\ldots,k$ may be eliminated from NI without altering the acceptability of NI with respect to Σ, stipulated by condition (1). To show this, note that NI - $\{h_u\}$ is acceptable with respect to Σ in case the contraction of z^t with respect to component u constitutes a mapping. Thus, conditions (1) and (2) are fulfilled if and only if NI is a minimal set acceptable with respect to Σ. Δ

The set of all prime neighborhood indices with respect to a given set of sequences Σ, called SNI, can be determined by a repeated application of Theorem 2.16. First, it is chosen a set NI acceptable with respect to Σ. Next, the prime neighborhood indices are obtained by eliminating components in NI sequentially, in such a manner that the sets produced by such deletions are also acceptable with respect to Σ.

The structural complexity of a RPN may be measured by the size of its neighborhood index NI and the cardinality of its set of admissible local functions I. When first priority is given to the minimization of the number of connection lines, an optimal neighborhood index NI is determined by choosing in set SNI an element of minimum cardinality.

If the criterion of structural complexity emphasizes the minimization of the cardinality of the set of "instructions" of the RPN, NI in SNI is optimal whenever the set $\{z^t \mid z^t(c_t(NI\cdot i)) = c_{t+1},\ i \in Q,\ t \geq 1\}$ originated by NI and Σ is of minimum cardinality. By minimizing a cost function of the form $a|NI| + (1-a)|I|$, $0 \leq a \leq 1$, where NI is in SNI and I is originated by NI and Σ, trade-offs between NI and I can be made explicit.

CHAPTER III

SOME COMPUTATIONAL PROPERTIES OF REGULAR PROCESSOR NETWORKS

In this chapter, first we give an alternative representation for the topology of a RPN, comparing it to the definition of Chapter 1.

3.1. Homogeneous Neighborhood Maps

The concept of a homogeneous neighborhood map is introduced in this section. It is then proved that the class of connected digraphs defined by homogeneous neighborhood maps coincides with the class of quasigroup graphs.

DEFINITION 3.1

Let D be an arc-labeled digraph. The *neighborhood map* N_D of D is a mapping N_D: $P(D) \to C(P(D))$, where $C(S)$ denotes the set of indexed subsets of a set S, such that $N_D(v) = \{u_1, u_2, .., u_\nu, ..\}$ specifies the set of points adjacent to point v, and arc (u_ν, v), $\nu \geq 1$, has label ν in D. Indexed set $N_D(v)$ is called the *neighborhood* of point v in D.

Let D be an arc-labeled digraph with point set P and neighborhood map N. Map N is called *homogeneous* in case

(1) For all points $v \in P$, $|N(v)| = k$, for some finite integer k.

(2) Maps $\beta_\nu N$, $\nu=1,2,..,k$ are bijective, where projection β_ν: $P^k \to P$ is defined by $\beta_\nu(n_1, n_2, .., n_k) = n_\nu$ for all $(n_1, n_2, .., n_k) \in P^k$.

THEOREM 3.1

The class of connected labeled digraphs defined by homogeneous neighborhood maps is identical to the class of quasigroup graphs.

PROOF. Let D be an arc-labeled connected digraph with homogeneous neighborhood map N. Since $|N(v)| = k$, then $id(v) = k$ for all $v \varepsilon P(D)$. Since maps $\beta_\nu N$, $\nu=1,2,..,k$ are bijective, also maps $[\beta_\nu N]^{-1}$ are bijective. Thus, D is distinctly labeled with arc-coloring λ such that $\lambda([\beta_\nu N](v),v) = \nu$, $\nu=1,2,..,k$.

Since $|N(v)| = k$ and $[\beta_{\nu_1} N](v) = [\beta_{\nu_2} N](v)$ implies $\nu_1 = \nu_2$, then $od(v) = k$ for all $v \varepsilon P(D)$. Hence D is an arc-colored quasiregular digraph.

Let $D_{Q,NI}$ be a quasigroup graph. By definition, $N(v) = NI \cdot v$ is its neighborhood map. Since Q is cancellative, $\beta_\nu N: Q \to Q$ is one-to-one and onto, $\nu=1,2,..,|NI|$. Thus, map N so specified is homogeneous. Δ

Theorem 3.1 provides us with an alternative definition for the interconnection digraph of a RPN.

A neighborhood map $N:P \to \mathcal{C}(P)$ on a set of points P specifies a labeled quasiregular digraph only if N is homogeneous. Checking the homogeneity of N involves testing the bijectivity of $\beta_\nu N$, $\nu \geq 1$, and verifying whether $|N(v)| = k$ for all v in P. Whenever the interconnection digraph of a RPN is highly regular, the quasigroup specification usually constitutes a relatively more compact way to define these digraphs. A discussion of the computational complexity of problems defined on group-graphs has been provided by Knuth and Bendix [56].

3.2. Computation with RPN's

Observations on the computational power of some classes of RPN's are made in this section. Specifically, we examine the performance of RPN's with respect to computations for which no algorithms are known whose worst case execution time on a Turing machine is bounded by a polynomial in the size of the input data.

First, one-dimensional arrays are considered and limitations inherent to this basic system are pointed out. RPN's whose interconnection digraphs are trees, are then investigated. Finally, a class of machines the size of whose neighborhood structure grows with the size of the input data is analyzed.

3.2.1. A Class of Combinatorial Problems

A way to analyze the inherent complexity of problems is to view them as recognition problems for formal languages. For example, consider the well known "traveling salesperson" problem. A network with n nodes is given, in which each arc has a specified "length." One is asked to determine a cycle of minimum length which contains each of the nodes of the network. A recognition problem that corresponds to the traveling salesperson problem is defined as follows. Given a string of symbols representing the network and an integer m, accept this string as a sentence of the language if the network contains a spanning cycle of length not greater than m; otherwise reject the string.

The reader is assumed to be familiar with the notion of a (deterministic) Turing machine. This machine is able to modify only a bounded amount of its storage on a single computational step. RPN's, which have an unbounded number of interconnected processors, are machines capable of unbounded parallelism. Thus, it is to be expected that RPN's should be computationally more powerful than machines that can not perform unbounded parallelism (see Section 1.2).

A *nondeterministic Turing machine* is like an ordinary (deterministic) Turing machine, except that state transitions are not necessarily uniquely determined. That is, given a state and list of tape symbols, the machine has a finite set of choices of state transition. An input string s is deemed accepted if at least one sequence of transitions with s as input leads to an accepting state. On a given input s, one can think of a nondeterministic Turing machine M as executing all possible sequences of transitions (or "moves") in parallel until either an accepting state is reached or no more moves are possible. It is convenient to think of M as "guessing" only an accepting sequence.

Let us define P as the family of all languages which can be accepted by deterministic Turing machines whose running time is bounded by a polynomial in the size of the input, and NP the corresponding family of languages which can be accepted by nondeterministic Turing machines in polynomial time. Many important problems which are not known to be in P are in NP. For a detailed discussion of these problems and a good bibliography, see Aho et al. [62].

There is strong evidence that the two classes may not be the same. It has been shown by Karp and Cook [62] that many problems in NP would be in P if and only if NP and P were identical. The equivalence class of problems in NP having this property is called *polynomial-complete*. Either all of these problems admit some algorithm executable in polynomial-time by a Turing machine or none of them does; no such algorithm is currently known.

The classes P and NP are invariant under a wide range of changes of machine model (see [62]). Polynomial-complete problems include the satisfiability problem of a Boolean expression in conjunctive normal form, traveling salesman, determining the maximum clique or minimal coloring of a graph, scheduling, register allocation, 0-1 integer programming.

An algorithm for a problem in NP can be regarded as a procedure which, when confronted with a choice between (say) two alternative transitions, can create two "copies" of itself, and follow up the consequences of both courses of action. Thus, the possible sequence of transitions that the machine can make on an input i can be arranged into a tree of descriptions. Each path from the root to a leaf in the tree represents a sequence of possible transitions. If s is a shortest sequence of transitions that terminates in an accepting state, then as soon as the machine has made this transition sequence s, the machine stops and accepts the input i. There is some constant k such that there are no more than k choices of next move in any situation. Each sequence of transitions leading to a halt of the nondeterministic Turing machine that executes the algorithm in NP is of polynomial length. Thus, a problem in NP can be computed on a deterministic Turing machine by tracing out a tree of depth bounded by a polynomial, thus taking exponential-time in the worst case.

Formally, a sequence of up to t(n) transitions of a nondeterministic Turing machine M_1, where n is the length of the input, is represented by a string over the alphabet AL = $\{0,1,\dots,k-1\}$ of length up to $t(n)$, where k is as above. A deterministic machine M_2 simulates M_1 on an input i of size n as follows. M_2 successively generates all strings v over alphabet AL of length at most $t(n)$ in lexicographic order. There are no more than $(k + 1)^{t(n)}$ such strings. As soon as a new string is generated, M_2 simulates s_v, the sequence of moves of M_1 represented by v. If s_v causes M_1 to halt (generating a solution), then M_2 also halts. If s_v does not represent a valid sequence of moves by M_1 or if s_v does not cause M_1 to accept i, then M_2 repeats the process with the next string over AL.

Machine M_2 can simulate sequence s_v in time up to $t(n)$. It takes at most time $t(n)$ to generate each string v. Therefore the entire simulation of M_1 by M_2 can take time up to $2t(n)(k+1)^{t(n)}$. However, there exists no proof that all methods for converting a nondeterministic algorithm to a deterministic one must require an exponential loss of time.

The idea of simulating the operation of a nondeterministic Turing machine with a RPN will be explored in the following sections. This idea follows fairly immediately from the definition, as the reader observed in the above discussion. The idea of performing all possible transitions in parallel has been used as a conceptual aid (see Aho et al. [62]) for understanding nondeterministic machines. From a detailed examination of the peculiar nature of RPN's, it will be possible to derive some not obvious trade-offs between time and hardware complexity.

In the above simulation of a nondeterministic Turing machine, all possible sequences of transitions on a given input i were arranged into a tree of states. All accepting states in the tree were considered equally desirable. This represents an exhaustive search which examines a number of states exponential in the size of input i. However, for many problems in class NP, one can formulate a search guided by the use of heuristic merit functions which will reduce the number of points in the tree that must be examined.

In some algorithms, points in the tree are generated in some order, which may depend on information that was found at points previously traversed, and a test is performed at each point to determine whether it constitutes a solution (i.e., an accepting state). In another variant, solutions may themselves be ordered by merit and the goal set of finding the best solution. Thus, there is a distinction between a search for an *all-or-none solution* and a search for a *best solution*. A problem where the first type of solution is required will be called *all-or-none*; a problem involving the latter type of solution will be termed *optimization problem*.

In a heuristic search, one provides a merit function for points in the tree that will recommend the most promising point from which to search next. Roughly, the idea is to examine the tree points to determine their likelihood of yielding a (best) solution, and to avoid traversing unpromising points. This heuristic approach has been used by Karp [54] and others to derive some polynomial-time Turing algorithms for optimization problems in class NP. These algorithms provide approximately best solutions for all inputs, or best solutions for all except a finite number of inputs, or approximately best solutions for all except a finite number of inputs. Algorithms with these properties will be called *approximation algorithms*.

3.2.2. Some Observations on Computational Complexity

The hardware complexity of RPN's that execute problems in class NP within polynomial-time will be explored in this section. It will be shown that a distinction between all-or-none and optimization problems must be made. Also, it will be shown that consideration must be given to the mechanism by which a solution is outputted from the machine.

DEFINITION 3.2

A RPN M with cellular space Q is said to be *output-constrained* in case there exists a special cell v in Q such that M assumes an accepting state configuration c whenever c(v) is an accepting state. Such a cell v is called an *output cell* of RPN M. If no such restriction exists, a RPN is called *output-unconstrained*. A RPN is said to accept a language on s(n)-*bounded-space* if the number of processors that intervene in the computation is not greater than s(n).

THEOREM 3.2

An all-or-none problem in class NP can be executed by an output-unconstrained RPN on exponentially-bounded-space within polynomial time.

PROOF. A parallel simulation of the nondeterministic Turing machine will be performed. Each processor in the RPN will execute a sequence or transitions s_v (defined in previous section) of the nondeterministic Turing machine on the given input i. In this manner, the RPN will trace out the tree of all possible sequences of transitions on i, simulating its operation. If s is a shortest sequence of transitions that terminates in an accepting state of the nondeterministic Turing machine, then the time used in processing input i will be the length of s. Note that to compute an optimization problem as above would take exponential time, since an exponential number of steps is needed to determine a solution of best merit.

Whenever one may formulate heuristic merit functions for a problem, the search tree will be "pruned" as the computation proceeds. Each processor will compare its information with that of processors polynomially distant from it. In that manner, cells pursuing unpromising searches can be eliminated from the computational process and for some problems, the average number of intervening processors significantly reduced (see Karp [54]). Δ

DEFINITION 3.3

An algorithm A is said to be *polynomial-space-aligned* with respect to an output-constrained RPN M whose output cell is v, in case all solutions generated by A on each input i, are produced in cells whose distance to cell v is bounded by a polynomial in the size of the problem.

COROLLARY 3.3

A problem in class NP which admits an algorithm that is polynomial-space-aligned with respect to an output-constrained RPN M can be computed by M on exponentially-bounded-space within polynomial time.

PROOF. If all cells that generate solutions (i.e., enter accepting states) are polynomially distant to output cell v, then a best solution may be selected within polynomial time after a process of comparison of intervening solution merit values. This selection of a maximum merit value can be performed while the solutions propagate, by inhibiting the propagation of solutions whose values are not optimal. Δ

Let a RPN M be modified, so that a cell v is incorporated to the network which has every other cell as a neighbor. Suppose this modified M is output-constrained with cell v as an output cell. Then the requirement of polynomial-space-alignment in Corollary 3.3 may be lifted, since it is satisfied automatically. For example, consider a one-dimensional linear-array with an origin cell directly connected to every other cell. Such a machine would require a circuit capable of sorting the solution merit values in polynomial-time. This may be done by performing a binary search (see Aho et al. [62]).

An algorithm which requires a resource that must grow exponentially can not be considered practical for large input sizes. Hartmanis and Simon [63] have proved that random access machines (RAM's) capable of computing the product of two operands in unit time, may solve problems in class NP within polynomial-time at the expense of an exponential amount of storage. Therefore it seems that the computation of problems in class NP can be speeded up asymptotically only at a prohibitive expense in terms of hardware complexity. However, since many of the problems in NP need to be solved (for example, the traveling salesperson problem), it is still of interest to attempt the generation of faster solutions in the input size range dictated by the applications.

Let us examine the speedup achievable with RPN's in the generation of approximate solutions to problems in class NP. An approximate Turing algorithm which traverses a polynomial number $p(n)$ of paths in the search tree, takes time up to $p(n)t(n)$ to solve a problem in NP, where $t(n)$ is the length of a longest sequence of transitions, and n the input size. A parallel simulation of this algorithm carried out on a RPN will take time up to $t(n)$ on $p(n)$-bounded-space, resulting in a speedup by a factor $p(n)$.

A RPN such that the path distance between every two points of its underlying digraph is bounded by a logarithm in the number of points will be examined in the next section.

3.3 Balanced-Tree RPN's

DEFINITION 3.4

A *tree* is a connected digraph with no cycles which has a unique path from one point, called the *root*, to every other point in the digraph. The *depth* of a point u in the tree is the length of the path from the root to u. A tree is *k-balanced*, if all points other than the root are of the same degree k + 1, $k \geq 2$.

The number of points to depth d of a k-balanced tree, $k \geq 2$, is $(k^{d+1} - 1)/(k - 1)$, which approaches k^d from above as k increases. Thus, the path distance between every two points in a balanced tree is bounded by a logarithm of the number of points in the tree.

THEOREM 3.4

Let a problem be solved within polynomial-time by a nondeterministic Turing machine M. Then the problem is solvable by an output-constrained balanced-tree-RPN on exponentially-bounded-space within polynomial time.

PROOF. The idea of the proof is again to simulate the operation of M. Each sequence of transition s of M, is executed by some cell in the balanced-tree-RPN. Since the path distance is bounded by a polynomial of the input size, every pair of processors may interchange information and a best solution be displayed at the root within polynomial time. The number of intervening processors may be reduced by the use of heuristic functions. Δ

It is not feasible to fabricate an arbitrary tree structure in physical space, so that the physical distance between adjacent points be the same throughout the tree. To see this, note that the number of points in a tree grows exponentially with path distance, whereas the number of points in a physical realization can only grow polynomially with physical distance. Three-dimensional realizations of 7, 25 and 50- balanced trees have been proposed by Lipovsky [57]. He has also provided upper bounds on the propagation delay through such physical realizations. Since an unbounded loss of physical proximity takes place between points in a realization of a tree, Theorem 3.4 is not applicable for large values of problem size (such that propagation delays are no longer negligible as compared to computation time).

3.4. Flexible Interconnection Structures

In this section, we consider d-dimensional tessellation processor networks (TPN's) whose dimension d is allowed to grow with the size of an input problem. In a common TPN, only a polynomial number of points are within mutual polynomial path distance. How the situation changes when the number of connections can grow with input size will be examined next.

THEOREM 3.5

An output-constrained d-dimensional TPN such that $d = n/\log p(n)$, where $p(n)$ is a polynomial in problem size n, can be used to solve problems in class NP on exponentially-bounded-tape within polynomial time.

PROOF. If the number of computing cells is b^n, for some constant b, then each edge of the d-dimensional cube that contains these cells must be of length $b^{n/d}$. The computational process performed by the TPN is a parallel simulation of the nondeterministic Turing algorithm for the given problem, as in previous sections. Every two cells in the cube of edge $b^{n/d}$ are within path distance $d^{1/2}b^{n/d}$. Therefore, if $d = n/\log_b p(n)$, then the maximum path distance becomes $(n/\log_b p(n))^{1/2} p(n)$. Δ

CHAPTER IV

DECENTRALIZED INTERPROCESSOR MESSAGE-ROUTING MECHANISMS FOR RPN's

4.1. Introduction

The implementation of parallel processing systems presents a number of challenges not found in single processor systems. Objectives that become crucial are hardware resource sharing, software resource sharing and, in the case of multiple-computer systems, load sharing; for a survey, see Enslow [58] and Wulf [59]. Hardware resource sharing takes place, for example, when a task executing on a particular processor uses a device located at a different processor. Software resource sharing includes the ability for a processor to obtain programs and data residing on another processor. In load sharing, the goal is to transfer jobs from one processor to another in order to achieve an efficient operation of the system. The levels at which interaction between processors must occur certainly vary from system to system, but the success of most designs will depend to a great extent on the adequacy with which interprocessor communication can take place.

The concern of this chapter is to characterize decentralized interprocessor message-routing schemes for RPN's, that do not require knowledge of processor sites, or the identity of senders and receivers. Such systems have the important advantage over those which do require processor site knowledge, that overhead due to establishing, moving and eliminating connections does not exist. Since these routing schemes are decentralized, the necessary control logic in an implementation can be distributed throughout the network, a desirable modular feature. Enslow [58] has indicated some potential benefits inherent to a multiprocessor operating system with a decentralized organization that treats processors as an anonymous pool of resources. We quote:

> Such symmetric operation provides graceful degradation; it can provide better availability of a reduced capacity system; it provides true redundancy; and it makes the most efficient use of resources available.

A multiple-computer system which has incorporated the concept of anonymous treatment of processors is the Distributed Computer System [15], DCS, built at the University of California, Irvine. The interconnection graph of DCS is a ring supporting transmission in one direction only. Processors are connected to the transmission lines through a piece of hardware called the "ring interface," which recognizes software-defined process names. More specifically, messages circulating in the ring, consisting of both control information and data, are addressed to logical processes, rather than physical processors. Since each message is explicitly addressed to a process, each process must have a unique name; therefore, a mechanism is provided for preventing name duplication. Load sharing in the DCS design is achieved by a "bidding" scheme. Prior to initiating a task, for example a compilation, a "request for bid" is broadcast to all processors. Those processors with both the ability and available capacity to execute the task respond. The bid is proportional to the excess capacity on the processor which responds. A "contract" is signed with the lowest bidder, and the task is executed on the winning processor.

Since the interconnection digraph of DCS is a ring, its control mechanism does not need to be concerned with message routing. In a general network with a large number of processors, however, factors such as distance, communication costs and bandwidth are likely to become determinant. It is then necessary to formulate some mechanisms, hopefully decentralized, by which messages can be routed between sender and receiver as directly as possible.

When there is a lack of knowledge about a task or environment, many times an organized heuristic search is the only way to solve a problem. By a heuristic, we mean any device that aids search effectiveness by either restricting the region searched or appropriately ordering the search. In those difficult cases, a decentralized control scheme that incorporates the "bidding" concept seen in the DCS design, may become useful. Let each processor in a network explore a particular hypothesis or strategy and relate to the other processors the validity of its approach. The control mechanism of the system would pass command to the processor that had progressed the most in its exploration (i.e., was the closest to the goal), or which had the most valuable information. The processor in command could then direct a cooperative effort in meeting the computational demands. More specifically, it would monitor the progress made by the other processors, interrogate them, discarding those heuristics that seemed of little value, and formulate new heuristics to replace the discarded ones.

The "bidding" concept may be utilized, for example, when extracting features in pictorial patterns. The computational goal then is to discover a set of features of subpatterns of the picture that are mutually compatible, and which will allow the inference of a global statement about the picture. Each processor would hypothesize a feature on a subpattern of the figure, and then compare the feature with the ones hypothesized by the other processors. The goal can be achieved after a process of information interchange, using the "bidding" notion described above. The detailed characteristics of the process will depend of course on the particular heuristic procedure that is applied.

Further examples of systems which would benefit from decentralized routing schemes that operate in the absence of processor site knowledge, discussed by Farber [15], include distributed data management and real-time command systems.

In systems that are message-oriented, like the ones discussed above, interprocessor communication will consist essentially of the broadcast of interrogating messages, followed by the subsequent routing of response messages to the broadcast senders. To generate decentralized response routing in the absence of knowledge of where a broadcast sender or its responding processors are located in the network, some information on how the broadcast message propagates through the network has to be extracted and exploited. The response routing schemes investigated in this chapter are those which are achieved by applying at each processor in the response route, a local routing rule that is a function of the set of input lines upon which the broadcast message first arrives to that processor and possibly the input line on which the response message reaches that processor; thus, the switching decision at each processor is based on strictly local information. It is assumed that the time it takes for a message to go from a processor to an adjacent one is the same at all processors, and we define this interval as *unit time*.

Four examples of nontrivial networks which can support message routing schemes not requiring information of processor sites, are due to Sahin [60]. The examples given (which may be viewed as RPN's) are two "square" and two "hexagonal" networks. No attempt to show the validity of the schemes or to indicate how a switching decision rule may be derived was made. To our knowledge, no other schemes ignoring identity of senders and receivers have been proposed.

Several fundamental questions concerning interprocessor site-knowledge-free message-routing schemes (called ISKF schemes) are resolved in this chapter. First, the class of RPN's which admit ISKF schemes with monotonic response-message convergence is characterized. For a RPN in that class, the generating local routing rule is given. A simple necessary condition for the existence of an arc-labeling in a given digraph, so that it lends itself to the definition of an ISKF scheme, is then established. A characterization of a local routing rule pertaining to a symmetric digraph is given.

Next, more general properties, pseudomonotonic and strong message-routing convergence, are introduced. Several conditions satisfied by digraphs that support these properties are proved. Finally, we give some measures of quality, useful in a comparative study of network topologies.

4.2. Interprocessor Message-Routing Schemes

First of all, the following graph-theoretic definitions are in order. For convenience, pertinent concepts introduced in Def. 1.1 will be repeated.

DEFINITION 4.1

The *adjacency matrix* AD of a digraph D = (P(D), A(D)) has an entry

$AD_{uv} = a_{uv}$ whenever the pair of points (u,v) is an arc in $A(D)$, and $AD_{uv} = 0$ otherwise. The *distance* $d(u,v)$ from point u to point v is the length of a shortest walk (or path) from u to v. The *path matrix* $PA(m)$ of a connected digraph is the matrix whose entry $PA_{uv}(m)$, if $m \geq d(u,v) = d$, equals the set of shortest paths from u to v, written as a sum of words on symbols $a_{uv} = AD_{uv}$ of the form $a_{ux_1} a_{x_1x_2} \cdots a_{x_{(d-1)}v} + \ldots + a_{uz_1} a_{z_1z_2} \cdots a_{z_{(d-1)}v}$; otherwise, $PA_{uv}(m) = 0$. The *shortest path matrix* SH of a connected digraph is defined to be $SH = \lim_{m \to \infty} PA(m)$.

LEMMA 4.1

The path matrix $PA(m)$, $m \geq 1$, is obtained from the adjacency matrix AD as follows (see Aho et. al. [62] for an alternative approach):

$$PA(1) = AD ,$$

$$PA_{uv}(m+1) = \begin{cases} \sum\limits_{e} PA_{ue}(m) \cdot AD_{ev} & \text{if } PA_{uv}(m) = 0, \\ PA_{uv}(m), & \text{otherwise.} \end{cases}$$

PROOF. The product $AD_{ue} AD_{ev}$ will be nonzero whenever there exists a walk $a_{ue} a_{ev}$ from u to v. Thus, AD^2_{uv} represents the set of walks of length 2 from u to v. It follows that $PA_{uv}(m) = 0$ whenever $m < d(u,v)$; otherwise, $PA_{uv}(m)$ as constructed above specifies the shortest paths from u to v. Δ

DEFINITION 4.2

A digraph D is *k-arc-colorable* if there exists an assignment of k colors to its arcs, so that no two arcs with a common initial or terminal point have the same color. A *1-factor* of D is a spanning subgraph of D which is quasiregular of degree 1. A quasiregular digraph D of degree k is k-colorable whenever D is the arc-disjoint union of k 1-factors, since the set of arcs in each 1-factor may be assigned a distinct color, and conversely. Thus, the concepts of arc-coloration and 1-factorization become equivalent when applied to the digraph of a RPN. The reader is reminded of the fact that if $D = D_{Q,NI}$ (the quasigroup graph of quasigroup Q), the arcs belonging to a common 1-factor are assigned a common color $h \varepsilon NI$. The *arc-labeling* λ of quasigroup-graph $D_{Q,NI}$ is the mapping λ: $A(D_{Q,NI}) \to NI$ such that $\lambda(u,v) = h$, $h \varepsilon NI$, whenever $u = hv$. Since by definition of a quasigroup, λ is an arc-coloring of $D_{Q,NI}$, if AD is the adjacency matrix of $D_{Q,NI}$, then $\lambda(AD_{uv_1}) = \lambda(AD_{uv_2}) \neq 0$ implies $v_1 = v_2$, for $u,v_1,v_2 \;\varepsilon\; Q = P(D_{Q,NI})$.

Theorem 1.4 proved that every quasiregular digraph D can be represented as a quasigroup-graph $D_{Q,NI}$ of some quasigroup Q, such that $Q = P(D)$. A presentation (NI,R) for Q is obtained from D by first expressing D as the arc-disjoint union of 1-factors; coloring the 1-factors with the symbols of NI; and finally deriving a set of defining relations R over NI as follows. An equality $r = s$ of positive formulas r,s will hold whenever r and s are spelled by two distinct walks w'_{uv} and w''_{uv} between points u and v, respectively.

A positive formula x over alphabet NI is said to be *canonical*, if there exists no other positive formula w equivalent to x under presentation (NI,R) that is of smaller length; this is written x = can(w). The leftmost symbol of a formula w is denoted lm(w); the rightmost symbol is denoted rm(w).

LEMMA 4.2

Let SH be the shortest path matrix of a quasigroup-graph $D_{Q,NI}$ with arc-labeling λ. Then $\lambda(SH_{uv}) = \sum_{w \in W_{uv}} can(w)$, $W_{uv} = \{w_1(..(w_{d-1}w_d)..) \mid u = w_1(..(w_d v).)\}$, $u, v \in Q$.

PROOF. A positive formula $w \in W_{uv}$ represents a walk from $u = w_1(..(w_d v).)$ to v. Thus, $\sum_{w \in W_{uv}} can(w)$ equals the sum of positive formulas spelled by shortest paths from u to v, i.e., $\lambda(SH_{uv})$. Δ

DEFINITION 4.3

The *first arrival matrix* FA of a connected digraph D is defined to have entries FA_{uv} equalling the set $\{a_{yv}\}$, where (y,v) is an arc included in a shortest path from u to v. The *exit matrix* EX of D has entries EX_{uv} equalling the set $\{a_{uy}\}$, where (u,y) is an arc included in a shortest path from u to v.

LEMMA 4.3

Let SH, FA and EX be the shortest path, first arrival and exit matrices of D, respectively. Then $FA_{uv} = \{rm(SH_{uv})\}$, $EX_{uv} = \{lm(SH_{uv})\}$, where $rm(w_1+..+w_d) = \{rm(w_1),..,rm(w_d)\}$, $lm(w_1+..+w_d) = \{lm(w_1),..,lm(w_d)\}$, $w_1+..+w_d \in SH_{uv}$.

PROOF. FA_{uv} denotes the set of last arcs and EX_{uv} the set of first arcs in shortest paths from u to v. The identities follow from the definition $\lim_{m \to \infty} PA(m) = SH$. Δ

From Lemmas 4.1 through 4.3 it is seen that the matrices $\lambda(FA)$ and

$\lambda(EX)$ of a digraph $D_{Q,NI}$ may be computed either from the adjacency matrix AD of $D_{Q,NI}$, or from a presentation (NI,R) of quasigroup Q. The first approach is more mechanical. However, if the set R of defining relations of Q is small, the second approach may require considerably less effort. The latter approach will be illustrated in Examples 4.1 and 4.2.

The following definitions formalize the notion of an ISKF scheme.

DEFINITION 4.4

An *elementary local routing rule* ψ on digraph $D_{Q,NI}$ with arc-labeling λ, is a mapping ψ: $2^{NI} \to 2^{NI}$. An *extended local routing rule* ϕ is a mapping $NI \times 2^{NI} \to 2^{NI}$. A *local rule* is either an elementary or an extended local routing rule. A local rule is *deterministic* if each of its values is a set of cardinality equal to one. A local rule ϕ is *nondeterministic at* X in its domain, if $|\phi(X)| \geq 1$.

Given a digraph $D_{Q\ ,NI}$ with arc-labeling λ and an arbitrary point s in it, the *transition function* T: $Q \to 2^Q$ *induced by elementary local routing rule* ψ with broadcast sender s, is the mapping specified by $\lambda(v,T(v))= \psi(\lambda(FA_{sv}))\neq\phi$, for $s\neq v\varepsilon Q$, $T(s) = \{s\}$. The *transition function induced by extended local routing rule* ϕ with broadcast sender s and respondent r, if the mapping T: $Q \to 2^Q$ specified by $\lambda(r,T(r))=\phi(h,\lambda(FA_{sr}))\neq \phi$, for arbitrary $h\varepsilon NI$, $r\neq s$; $\lambda(v,T(v))= \phi(\lambda(y,v),\lambda(FA_{sv}))$, where $v\neq s$, $v\varepsilon T^n(r)$, $y\varepsilon T^{n-1}(r)$, $n\geq 1$, (y,v) is an arc of $D_{Q,NI}$; and $T(s) = \{s\}$.

A local routing rule ϕ on a given digraph $D_{Q,NI}$ is *response-message convergent on* $D_{Q,NI}$ if the transition function T induced by ϕ satisfies $T^n(r) = \{s,..,s\}$, for all points s,r of $D_{Q,NI}$ and some finite integer $n \geq 1$. If a local routing rule ϕ is response-message convergent on $D_{Q,NI}$, we say that ϕ *generates an interprocessor site-knowledge-free message-routing scheme* (ISKF *scheme*) on $D_{Q,NI}$. More informally, a local rule ϕ generates an ISKF scheme on $D_{Q,NI}$ whenever homing of response messages to a broadcast sender results from routing at each point as prescribed by ϕ. If ϕ is an elementary local routing rule, its value is a function of the set of input lines upon which the broadcast message first arrives (by Definition 4.3). If ϕ is an extended local routing rule, its value depends on the input line upon which the response message arrives, in addition to the first arrival pattern of the broadcast message.

If a local rule is nondeterministic on a large subset of its domain, then at many points in the digraph, there exists a choice as to which line to switch the response message to. Thus, the routing decision at those processors may include other considerations, such as congestion, reliability, etc., without altering the convergence of the ISKF scheme.

The ISKF schemes that present the fastest response convergence are those which achieve routing of response messages r to a broadcast senders along shortest paths. These are called *monotonic* ISKF *schemes*, since in them the trajectories $T^n(r)$, $n \geq 1$, of a response message r satisfy $d(T^{n+1}(r),s) < d(T^n(r), s) \neq 0$, i.e., their convergence is monotonic.

Clearly, monotonic ISKF schemes deserve a special interest. The following theorem characterizes them.

THEOREM 4.4

Let FA and EX be the first arrival and exit matrices of a digraph $D_{Q,NI}$ with arc-labeling λ. There exists an elementary local routing rule $\bar{\psi}$ that generates a monotonic ISKF scheme on $D_{Q,NI}$ if and only if

$\lambda(FA_{sv_1}) = \lambda(FA_{sv_2})$ implies $\lambda(EX_{v_1s}) \cap \lambda(EX_{v_2s}) \neq \Phi$ for all $s, v_1, v_2 \in Q$. If it exists, ψ satisfies $\psi(\lambda(FA_{sv})) \subseteq \lambda(EX_{vs})$, for all s, $v \in Q$.

PROOF. Local rule ψ exists whenever the transition function T, with broadcast sender s, induced by ψ satisfies $\lambda(v,T(v)) \subseteq \lambda(EX_{vs})$, for all $s \neq v \in Q$, since routing should proceed along shortest paths. By definition of T, $\lambda(v,T(v)) = \psi(\lambda(FA_{sv}))$, $v \neq s \in Q$. Thus, ψ exists whenever $\psi(\lambda(FA_{sv})) \subseteq \lambda(EX_{vs})$ is satisfied for all $v \neq s \in Q$. Since ψ is a function, and since $A \subseteq B_1$, $A \subseteq B_2$ imply $A \subseteq B_1 \cap B_2$, $\lambda(FA_{sv_1}) = \lambda(FA_{sv_2})$ implies $\psi(\lambda(FA_{sv_1})) = \psi(\lambda(FA_{sv_2})) \subseteq \lambda(EX_{v_1s}) \cap \lambda(EX_{v_2s})$, for all $v_1, v_2 \neq s \in Q$. Since Φ is not in the range of ψ, the proof is complete. Δ

The theorem and the definitions are illustrated in the following examples.

EXAMPLE 4.1

The digraph $D_{Q,NI}$ with group presentation (NI,R), NI = $\{a_1,..,a_d,a_1^{-1},..,a_d^{-1}\}$, $d \geq 1$, R = $\{gh = hg,\ h \neq g \in NI\}$, admits an ISKF scheme generated by an elementary local routing rule. The topology of $D_{Q,NI}$ is called von Neumann in the literature (see Definition 1.7.)

PROOF. The adjacency matrix AD is specified by

$$AD_{uv} = \begin{cases} a_{uv} & \text{if } u = hv,\ h \in NI,\ u,v \in Q, \\ 0 & \text{otherwise.} \end{cases}$$

Since the group operation is commutative, $\lambda(SH_{uv})$ = sum of formulas that permute the symbols of can(w), $u = wv$; $\lambda(FA_{uv}) = \lambda(EX_{uv}) = \{h \mid h$ is a symbol in can(w), $u = wv\}$; $\lambda(EX_{vu}) = [\lambda(EX_{uv})]^{-1} = \{h^{-1} \mid h \in \lambda(EX_{uv})\}$. Thus, elementary local routing rule ψ, $\psi(X) = X^{-1}$, $\Phi \neq X \subseteq NI$, generates a monotonic ISKF scheme on $D_{Q,NI}$. Δ

EXAMPLE 4.2

The digraph $D_{Q,NI}$ with group presentation (NI,R), NI = $\{a_1,a_2,a_3\}$, R = $\{gh = hg,\ g \neq h \in NI,\ a_1a_2a_3 = e\}$ admits an ISKF scheme generated by an elementary local routing rule.

PROOF. By commutativity, $\lambda(SH_{uv})$ = sum of permutations of can(w), u = wv. Every canonical positive formula can(w) is of the form can(w) = $g^q h^r$, for $g \neq h \in NI$ and $0 \leq q \leq r > 0$. To see this, observe that every word w is equivalent to an expression of the form $f^x g^y h^z$, for integers $x \leq y \leq z$ and $f,g,h \in NI$. Then $f^x g^y h^z = (fgh)^x g^{y-x} h^{z-x} = g^{y-x} h^{z-x}$, for mutually distinct $f,g,h \in NI$, which is in the above form.

Without loss of generality, consider now the case u = wv, where $w = a_2^q a_3^r$, $0 \leq q \leq r > 0$.

$$\lambda(EX_{uv}) = \lambda(FA_{uv}) = \begin{cases} \{a_2,a_3\} & \text{if } q \neq 0, \\ \{a_3\} & \text{if } q=0 \end{cases}$$

$w^{-1} = a_3^{-r} a_2^{-q} = (a_1a_2)^r a_2^{-q} = a_1^r a_2^{r-q}$, a canonical form.

Thus,

$$\lambda(EX_{vu}) = \begin{cases} \{a_1,a_2\} & \text{if } q \neq r, \\ \{a_1\} & \text{if } q=r. \end{cases}$$

Hence, for all u, v $\in$ Q, $\lambda(FA_{uv}) = \begin{cases} \{a_2,a_3\} \text{ implies } \{a_1\} \subseteq \lambda(EX_{vu}), \\ \{a_3\} \text{ implies } \{a_1,a_2\} = \lambda(EX_{vu}), \end{cases}$

since $q \neq r$ when $q = 0$.

Therefore, the elementary local routing rule ψ, where $\psi(\{f\})$ = $\{g,h\}$, $\psi(\{g,h\}) = \{f\}$, for mutually distinct $f,g,h \in NI$, generates a monotonic ISKF scheme on $D_{Q,NI}$.

The logic of the proof may be used to prove the following more general result. If $Q = (NI,R)$, where $NI = \{a_1,..,a_k\}$, $R = \{gh = hg,\ g \neq h \in NI,\ a_1a_2..a_k = e\}$, $k \geq 2$, then elementary local routing rule ψ, such that $\psi(X) = NI - X$, $\Phi \neq X \subseteq NI$, generates a monotonic ISKF scheme on $D_{Q,NI}$. Δ

Consider now the following question. Given a quasiregular digraph D, is there an arc-coloring λ of D such that there exists an elementary local routing rule which generates a monotonic ISKF scheme on D, with arc-coloring λ? The following theorem gives a necessary condition for a digraph D to lend itself to the definition of such a routing scheme.

THEOREM 4.5

Let FA and EX be the first arrival and exit matrices of a digraph D. There exists an arc-coloring λ of D and elementary local routing rule ψ that generates a monotonic ISKF scheme on D, with arc-coloring λ, only if $\bigcap_{(v,s)\in V} \lambda(EX_{vs}) \neq \Phi$, where $V = \{(s,v) \mid |FA_{sv}| = |NI|\}$.

PROOF. Pairs $(s,v) \in V$ satisfy $\lambda(FA_{sv}) = NI$, for all arc-colorings λ. By Theorem 4.4, (s,v_1), $(s,v_2) \in V$ imply $\lambda(EX_{v_1s}) \cap \lambda(EX_{v_2s}) \neq \Phi$; the theorem follows. Δ

The conditions of either Theorem 4.4 or 4.5, together with the property $\lambda(AD_{uv_1}) = \lambda(AD_{uv_2}) \neq 0$ implies $v_1 = v_2$, $\lambda(AD_{u_1v}) = \lambda(AD_{u_2v}) \neq 0$ implies $u_1 = u_2$ of an arc-coloring λ, can be used to prove the existence or inexistence of a monotonic ISKF scheme on a given digraph D. Inexistence is established whenever the resulting system of equalities becomes inconsistent. The impossibility of such a scheme on the "square-X" and "square-Y" digraphs studied by Sahin [60] was shown by the author in this fashion. The details are straightforward, and will be omitted.

Because of the special regularity of their topology, those digraphs $D_{Q,NI}$ that admit adjacency-preserving automorphisms, such that each automorphism corresponds to a permutation of the labels of the arcs of $D_{Q,NI}$, will deserve particular consideration. These digraphs will be called symmetric. Clearly, the 1-factors of a symmetric digraph are mutually isomorphic. A more formal definition is given next.

DEFINITION 4.5

A digraph $D_{Q,NI}$ is *symmetric*, whenever for every permutation $\pi:NI \to NI$, there exists a bijective mapping $\pi^*:Q \to Q$ such that $\pi^*(hv) = \pi^*(h)\pi^*(v), \pi^*(h) = \pi(h)$, for all $h \varepsilon NI$, $v \varepsilon Q$. An equivalent condition is $\lambda(AD_{\pi^*(u)\pi^*(v)}) = \pi(\lambda(AD_{uv})) \neq 0$, $u,v \varepsilon Q$, where λ is the arc-labeling of $D_{Q,NI}$, by definition of adjacency matrix AD.

The following theorem tells us how to test the symmetry of a digraph $D_{Q,NI}$, given a presentation of quasigroup Q.

THEOREM 4.6

A digraph $D_{Q,NI}$, with presentation $Q = (NI,R)$ is symmetric if and only if $Q = (NI,\pi^*(R))$, for all permutations $\pi:NI \to NI$, where $\pi^*(R) = \{\pi^*(r) \mid \text{equality } r \varepsilon R;\ \pi^*(hv) = \pi^*(h)\pi^*(v), \pi^*(h) = \pi(h),\ h \varepsilon NI,\ v \varepsilon Q\}$.

PROOF. Sufficiency: Suppose $Q = (NI,R) = (NI,\pi(R))$, for some permutation π, and consider mapping $\pi^*:Q \to Q$, defined as above. Then, since $Q = (NI,R) = (NI,\pi^*(R)$, an equality of positive formulas $v_1 = v_2$ is derivable from R whenever equality $\pi^*(v_1) = \pi^*(v_2)$ is derivable from R. Thus, mapping π^* is one-to-one. Since $(\pi^*)^{-1} = (\pi^{-1})^*$ has domain Q, π^*

is bijective. Finally, since $\pi^*(hv) = \pi(h)\ \pi^*(v)$, $h \in NI$, $v \in Q$, π^* is an automorphism.

Necessity: Let $D_{Q,NI}$, with $Q = (NI,R)$, be symmetric. By definition, for all permutations π and corresponding automorphisms π^*, equality of positive formulas $v_1 = v_2$ hold whenever equality $\pi^*(v_1) = \pi^*(v_2)$ does. Thus, sets of relations R and $\pi^*(R)$ are equivalent. Δ

An elementary local routing rule ψ that generates a monotonic ISKF scheme on a symmetric digraph $D_{Q,NI}$ is characterized in the following theorem.

THEOREM 4.7

Let there be an elementary local routing rule ψ that generates a monotonic ISKF scheme on a symmetric digraph $D_{Q,NI}$. Then $\psi(\pi(X)) = \pi(\psi(X))$, for each permutation $\pi:NI \to NI$ and $\Phi \neq X \subseteq NI$.

PROOF. By Theorem 4.4, $\psi(X) = \bigcap_{s,v \in V_X \neq \Phi} \lambda(EX_{vs})$, where $V_X = \{(s,v) | \lambda(FA_{sv}) = X\}$, $\Phi \neq X \subseteq NI$. Since by last theorem, equality of positive formulas $w_1 = w_2$ holds whenever equality $\pi^*(w_1) = \pi^*(w_2)$ does, $\pi^*(\lambda(SH_{uv})) = \lambda(SH_{\pi(u),\pi(v)})$, for all $u,v \in Q$, by Lemma 4.2. Hence, since $\pi(\lambda(FA_{uv})) = \lambda(FA_{\pi^*(u),\pi^*(v)})$, $\pi(V_X) = V_{\pi(X)}$. Thus, since $\lambda(EX_{\pi^*(u),\pi^*(v)}) = \pi(\lambda(EX_{uv}))$, $\psi(\pi(X)) = \bigcap_{s,v \in V_X \neq \Phi} \pi(\lambda(EX_{vs}))$. Because π is a permutation over NI, $\pi(A) \cap \pi(B) = \pi(A \cap B)$, for arbitrary sets $A, B \subseteq NI$. Thus, $\psi(\pi(X)) = \pi(\psi(X))$. Δ

An example of a symmetric digraph $D_{Q,NI}$, whose defining quasigroup Q is not a group, is given in Figure 1.1 of Chapter I.

Next, the notion of a *pseudomonotonic* ISKF *scheme* is introduced. Routing of response messages to the broadcast sender s, in such a scheme, is achieved so that the distance of the message to s decreases at every point in the route, with the possible exception of one point. An ISKF *scheme* is *strong*, if the distance of the message to s goes through a sequence of monotonically decreasing minima. A more formal definition of these notions is given next.

DEFINITION 4.6

A local routing rule ϕ is *anomalous* at pair s, x ε Q, if $d(T(x), s) \geq d(x,s)$, where T is the transition function with broadcast sender s induced by ϕ. Local rule ϕ generates a *pseudomonotonic* ISKF *scheme* on a digraph $D_{Q,NI}$, if when ϕ is anomalous at pair s,z, then ϕ is not anomalous at pair $s,T^n(z)$, $n \geq 1$. Clearly, such a ϕ is response message convergent, since the distance $d(T^n(r),s)$ of a message r to the sender s will decrease monotonically with time n, with the possible exception of one instant n, when ϕ is anomalous at $s,T^n(r)$. A local rule ϕ generates a *strong* ISKF *scheme* on a digraph $D_{Q,NI}$ whenever the integer function $\eta(n) = d(T^n(r),s)$, $n \geq 1$ is such that if $\eta(n_1 + 1) \geq \eta(n_1)$, $\eta(n_2 + 1) \geq \eta(n_2)$, $n_1 < n_2$, then $\eta(n_2) < \eta(n_1)$, $r \neq s \in Q$. Such a ϕ is response message convergent, since η has a sequence of monotonically decreasing minima.

THEOREM 4.8

A necessary and sufficient condition for the existence of an elementary local routing rule ψ generating a pseudomonotonic ISKF scheme on a digraph $D_{Q,NI}$, is the existence of a set $\Phi \neq NAN \subseteq Q^2$, such that:
(1) $\lambda(FA_{sv_1}) = \lambda(FA_{sv_2})$ implies $\lambda(EX_{v_1 s}) \cap \lambda(EX_{v_2 s}) \neq \Phi$, for (s,v_1), (s,v_2) ε NAN; and (2) $(s,T^n(z))$ ε NAN if $(s,z) \notin$ NAN, $n \geq 1$, where T

is the transition function with sender s induced by ψ, $\psi(X) = \bigcap_{s \neq v \varepsilon V_X \neq \Phi} \lambda(EX_{vs})$,

$V_X = \{(s,v) \varepsilon NAN | \lambda(FA_{sv}) = X\}$, $\Phi \neq X \subseteq NI$.

PROOF. By definition of the scheme, a generating rule ψ exists whenever if ψ is anomalous at (s,z), then ψ is not anomalous at $(s,T^n(z))$, $n \geq 1$. A rule ψ is not anomalous at pair (s,v) whenever $\psi(\lambda(FA_{sv})) \subseteq \lambda(EX_{vs})$; denote a set of such pairs by NAN. By Theorem 4.4, NAN must satisfy part (1) of the condition of the theorem. Consider the local rule ψ given by $\psi(X) = \bigcap_{s \neq v \varepsilon V_X \neq \Phi} \lambda(EX_{vs})$, $V_X = \{(s,v) \varepsilon NAN | \lambda(FA_{sv}) = X\}$, $\Phi \neq X \subseteq NI$. This rule exists if and only if $NAN \neq \Phi$. Thus, $(s,T^n(z)) \varepsilon NAN$ if $(s,z) \notin NAN$, $n \geq 1$, means that if ψ is anomalous at (s,z), then it is not anomalous at $(s,T^n(z))$, and the proof is complete. Δ

The following example is illustrative.

EXAMPLE 4.3

The digraph $D_{Q,NI}$ with group presentation (NI,R), where NI = {a,b}, $R = \{(ab)^2 = (a^{-1}b)^2 = e\}$, admits a strong ISKF scheme generated by elementary local routing rule ψ given by $\psi(\{a\}) = \{b\}$, $\psi(\{b\}) = \{a\}$, $\psi(\{a,b\}) = \{a\}$.

PROOF. The digraph is symmetric, since $(ba)^2 = (b^{-1}a)^2 = e$. Local rule ψ is anomalous only at (s,ws), w ε W, where W is the set of words of the form b^{2n} or $b^{2n}a$, $n \geq 1$. This is proved by showing that $\psi(\lambda(FA_{sv})) \subseteq \lambda(EX_{vs})$, whenever v ε (Q-W)s. The details are analogous to those in Example 4.2, and are omitted. If $w = b^{2n}$ or $b^{2n}a$, $T^6(ws) = b^{2(n-1)}$ or $b^{2(n-1)}a$, respectively, where $b^0 = e$. Thus, rule ψ does not generate a pseudomonotonic ISKF scheme on $D_{Q,NI}$, since ψ is anomalous at pairs (s,wz) and

$(s,T^6(wz))$. Since $d(b^{2n}s,s)=2n$ and $d(b^{2n}as,s)=2n+1$, $d(T^6(ws),s) <$ $d(ws,s), w \varepsilon W$; hence, ψ generates a strong scheme. Δ

A sufficient condition for the existence of pseudomonotonic ISKF schemes generated by extended local routing rules is provided by the following theorem.

THEOREM 4.9

Given a RPN with arc-coloring λ and digraph $D_{Q,NI}$, a sufficient condition for the existence of a pseudomonotonic ISKF scheme generated by an extended local routing rule ϕ, is (1) $\lambda(FA_{sv_1}) = \lambda(FA_{sv_2})$, and $H_{sv_1} \cap H_{sv_2} \neq \Phi \Rightarrow \lambda(EX_{v_1 s}) \cap \lambda(EX_{v_2 s}) \neq \Phi$, where

$H_{sv} = \{\lambda(y,v) | \lambda(y,v) \varepsilon \lambda(EX_{ys})\}$, y, v, s ε Q and (2) $\lambda(r,u) \varepsilon \phi(h,\lambda(FA_{sr})) \not\subseteq$ $\lambda(EX_{rs})$ for some h ε NI $\Rightarrow$ $\phi(\lambda(r,u), \lambda(FA_{su})) \subseteq \lambda(EX_{us})$.

PROOF. Let the condition be true. Then by (1) there exists an extended local routing rule ϕ specified by $\phi(h,X) = \bigcap_{s,v \varepsilon V_{h,X} \neq \Phi} \lambda(EX_{vs})$, where $V_{h,X} =$ $\{(s,v) | \lambda(FA_{sv}) = X,\ h \ \varepsilon\ H_{sv}\}$, $\Phi \neq X \subseteq NI$. If $d(T(r),s) < d(r,s)$, where r is the responding cell, the function T induced by this rule ϕ produces response routings such that $d(T^{n+1}(r),s) < d(T^n(r),s)$, $n \geq 1$, since in this case $\lambda(r,T(r)) \varepsilon H_{s,T(r)}$; this, in turn, implies $\phi(H_{s,T^n(r)}, \lambda(FA_{s,T^n(r)})) \subseteq \lambda(EX_{T^n(r),s})$, $n \geq 1$, by definition of ϕ and Theorem 4.4. Thus, routings proceed then along shortest paths.

Otherwise, when for some h ε NI $\phi(h,\lambda(FA_{sr})) \not\subseteq \lambda(EX_{rs})$, by (2) a transition $(u,T(u))$ from point $u = T(r)$ will satisfy $d(T(u),s) < d(u,s)$; therefore since $\lambda(u,T(u)) \varepsilon \lambda(EX_{us})$ then $\lambda(u, T(u)) \varepsilon H_{s,T(u)}$.

Thus whether or not the initial transition $(r,T(r))$ brings a response message closer to the broadcast sender, all subsequent transitions will. Δ

Some observations on the computational complexity of the conditions introduced in this chapter are in order. The algorithm provided in Lemma 4.1 requires $O(n^3)$ scalar operations to determine the shortest path matrix SH of a digraph D of n points. No presently known method for the all-pairs shortest path problem takes less than kn^3 time, for some constant k [62]. Also, $O(n^3)$ operations are needed to check the condition of Theorem 4.4 in matrices FA and EX obtained from SH. Thus, $O(n^3)$ time is required to determine whether a given labeled digraph admits a monotonic ISKF scheme if the adjacency matrix representation of the graph is used.

Alternatively, if the digraph representation in terms of generators and defining relations is utilized, the formula $\lambda(SH_{uv}) = \sum_{w \in W_{uv}} can(w)$ of Lemma 4.2 is applied. The time required to compute this formula will depend on the complexity of determining whether two formulas are equivalent under quasigroup Q. This complexity has been investigated among others by Knuth and Bendix [56]. Whenever a digraph has a highly regular structure, the condition of Theorem 4.4 can be checked in less than $O(n^3)$ time, as was illustrated by Examples 4.1 and 4.2.

In order to evaluate the routing performance of the schemes introduced in this chapter, it is of interest to define some measures of quality.

DEFINITION 4.7

The delay index d_P with respect to region $P \subseteq Q$ of an ISKF scheme whose transition function is T, is given by $d_P = \sum_{s,v \in P} (d(s,v) + r_{sv})$, where

r_{sv} is the smallest integer such that $T^{r_{sv}}(v) = \{s,\ldots,s\}$. The first summation $b_P = \sum_{s,v\in P} d(s,v)$ will be called the broadcast index; the second summation $r_P = \sum_{s,v\in P} r_{sv}$ will be termed the response index. Thus, $d_P = b_P + r_P$. Delay index d_P measures the average time needed by an arbitrary pair of processors to complete an information exchange.

A worst-case delay index wd_P with respect to region P is specified by $wd = \sum_{v\in P} (d(s,v) + r_{sv})$, where $P = \{v \mid d(s,v) \leq t\}$. Worst-case broadcast index wb_P and worst-case response index wr_P are defined as expected. The worst-case delay index wd_P provides a more simplified measure of the quality of a scheme. The following example compares a number of networks.

EXAMPLE 4.4

We shall evaluate the worst-case delay index wd_P, with $P = NI^{-t}$, for the following digraphs $D_{Q,NI}$: (1) d-dimensional Moore digraph (see Def. 1.7); (2) d-dimensional von Neumann digraph (see Example 4.1); (3) $NI=\{a,b\}$, $R=\{(ab)^2=(a^{-1}b)^2=e\}$; (4) $NI=\{a_1,a_2,a_3\}$, $R=\{gh=hg,\ g\neq h\in NI,\ a_1a_2a_3=e\}$;

Let $|P|=n$. Basic geometric relations will be used to compute $n(t)$.

(1) $n = t^d$; $wb_P = n^{1/d} = wr_P$; $wd_P = 2n^{1/d}$.

(2) $n = dt^d$; $wb_P = (n/d)^{1/d} = wr_P$; $wd_P = 2(n/d)^{1/d}$.

(3) $n = O(2t^2)$; $wb_P = (n/2)^{1/2} = wr_P$; $wd_P = (2n)^{1/2}$

(4) $n = O(1.5t^2)$; $wb_P = (2/3)^{1/2}n^{1/2}$; $wr_P = \frac{1}{2}(1+2)wb_P = 1.5wb_P$;

$wd_P = 2.5wb_P = 2.5(2/3)^{1/2}n^{1/2}$.

It appears that (3) is the most efficient network; although its complexity is the lowest (two input lines per processor), its routing performance is not lower than that of another network of the same dimension.

Other criteria for evaluating the topology of a network, in particular with respect to traffic capacity and reliability (the ability to function after a failure), are presented in the book by Davies and Barber [61, Chapter 12].

CHAPTER V

CONCLUDING REMARKS

The study of parallel processing systems interconnected in a regular manner is motivated both by considerations of practical adequacy and of analytic tractability, as explained in the Introduction.

With the availability of the microprocessor, it is now economically attractive to undertake the construction of large-scale parallel processing systems composed of small and inexpensive processors. Wulf [59] has stated: "I find the economic arguments in favor of aggregated small computers so compelling that I believe firmly that most future systems will be constructed in this way."

The main goal of this work has been to attempt to characterize the influence of the topologic structure of a regularly interconnected parallel processing system on some of its basic computational properties. A simplified mathematical setting, called RPN, was chosen for this purpose. Properties of the state transition digraphs of RPN's and also decentralized mechanisms by which processors can interchange information were studied.

The RPN model is meant to represent both "tightly coupled" systems, such as array processors, and "loosely coupled" systems, such as computer networks. A single operating system could control the entire network, or each of several interconnected separate computers may have its own largely autonomous operating system. Also, the level at which load and resource sharing take place in the network is immaterial to the model.

The topology of a RPN is specified by a finitely connected quasiregular directed graph. If the number of connection lines in the network is significantly smaller than the number of processors, as usually occurs in practice, it becomes inefficient to use a graph representation based on its adjacency matrix. A better representation for a directed graph, discussed in Chapter III, is then by means of its neighborhood list. If, in addition, the digraph is of a highly regular structure, it can sometimes be concisely represented by means of a quasigroup presentation in terms of generators and defining relations; for an illustration, the reader can see Examples 4.1 and 4.2. In Chapter I, pertinent properties of the quasigroup-graph formulation were given. Since some results existed in the literature for the case of finite digraphs only, proofs were provided for the infinite case.

Specifically, it was proved that a finitely connected quasiregular digraph is isomorphic to a quasigroup-graph $D_{Q,H}$, specified by some quasigroup Q and some finite subset H of Q. It was shown also that a special kind of quasiregular digraph, called regular, is isomorphic to a group-graph $D_{G,H}$, specified by some group G and some finite subset of H of G. As a result, the interconnection structure of a RPN (A,Q,NI,I) is describable by the set of generators NI of quasigroup Q. More precisely, set $NI \cdot i$ specifies the cells adjacent to cell i. Thus, neighborhood index NI is constant throughout the cellular space.

The characterization of the state transition graph of a uniform RPN was undertaken in Chapter II. First, a necessary and sufficient condition for the existence of transmitters in a general state transition digraph was provided. Fundamental to the proof of the condition are the notions of "properness" and "admissibility" of a tiling of the cellular space. A general method for the examination of basic properties of the state transition digraph was developed next. Questions such as connectedness, existence of unreachable states and maximum distance between given classes of finite states can be resolved with the technique. Several kinds of constraints on the structure or operation of a RPN may be added to the method. The method is based on a propositional language which expresses the transitional constraints of a particular RPN, and involves testing the solvability of a Boolean equation. A recursive characterization of this equation was provided.

A lower bound on the time required to reach from a given state configuration c_1, another arbitrary state configuration c_2, was proved in Section 2.3. It was demonstrated that this lower bound relates to the size of the largest periodic fragment in state c_1. Thus, it was established that the size of a periodic fragment in a state configuration provides a measure for the generating potential of that state.

A method for obtaining lower bounds on the structural complexity of uniform RPN's which generate a given set of state configuration sequences was presented at the end of Chapter II. The condition of Theorem 2.16 was used in the method to determine optimal neighborhood indices and instruction sets.

Several trade-offs between structural complexity and time of computation in RPN's were discussed in Chapter III. These trade-offs were analyzed in relation to the class of problems solvable within polynomial time by nondeterministic Turing machines. It was shown that a distinction must be made between problems which require some solution and those that require an optimal solution. In particular, RPN's of linear, balanced-tree and flexible interconnection topologies were investigated. It was demonstrated that the information input and output mechanisms play a role in a

RPN's computational capabilities.

Specifically, it was observed that all-or-none and polynomial space aligned problems in the above mentioned class can be executed within polynomial time by output-unconstrained and output-constrained RPN's, respectively.

Structural conditions under which an RPN admits the formulation of decentralized information routing schemes that do not require knowledge of processor sites were derived in Chapter IV. The conditions are based on the quasigroup-graph formulation of the interconnection topology of a network. Several categories of routing behavior, namely monotonic, pseudomonotonic and strong convergence were studied. Some examples which illustrate the application of the above techniques were provided.

In particular, a characterization of elementary and extended rules associated with the above routing schemes was provided. A property of elementary local rules pertaining to symmetric digraphs was established. Finally, some measures of routing performance were given which should be of value in a comparative study of network topologies. These decentralized information routing schemes may allow several information interchanges among processors to take place simultaneously.

The work carried out in this dissertation has uncovered a number of valuable allied problems. Some of our suggestions for future research will follow directly from the above results; other recommendations will not be so clearly connected to our investigation.

A method for examining properties of the transition digraph of a RPN was introduced in Chapter II. The identification of particular categories of RPN's admitting lower bounds on the size of their hypothetical unreachable partial state configurations would be of value in improving this method. The determination of additional lower bounds on the distance between an arbitrary pair of state configurations in a connected transition digraph would be useful as well in reducing the execution time of this procedure. The proofs of such results would probably be based in combinatorial arguments similar to that given in Section 2.1 for characterizing the existence of transmitter state configurations.

The mode of operation of a parallel processing system can be either SIMD (uniform) or MIMD (non-uniform), depending on whether or not the same "instruction" is executed at a given time on every processor of the system. Array processors such as ILLIAC IV are of the first type. It was demonstrated in Chapter II that uniformity in a RPN restricts computational speed. On the other hand, the introduction of non-uniformity generates a higher system complexity. By developing a categorization of processing environments for which the MIMD mode should be preferable over the SIMD mode, or vice versa, an aid in system design would be provided.

It seems natural to investigate the validity of the properties derived in this dissertation for mathematical models which are extensions of a RPN. In particular, asynchronic and probablistic models should be studied. Two choices come to mind when specifying asynchronous computations. The clocked option which allows a processor, at a discrete moment in time, not to perform a transition; and the speed independent option in which the transition instants are known to be within some bounds. The importance of incorporating adequate external input and output structures into a parallel processing device was pointed out by the results of Chapter III.

Another extension of the RPN model that seems of value, mentioned in Chapter II, is the partitioning of the set of admissible local maps into two disjoint subsets. One subset would include maps that process strictly local cell information. The other subset would be composed of maps that simply transfer information from one processor to another. This refinement of the model would allow to distinguish processing constraints from routing requirements in the structure of a RPN.

The notion of an admissible tiling in an infinite digraph was introduced in Section 2.1. A closer structural characterization of the class of digraphs which admit admissible tilings would simplify the statement of Theorem 2.6.

BIBLIOGRAPHY

1. STONE, H. S. Computer architecture in the 1980's. In *Computer Science and Scientific Computing*, J. M. Ortega, Ed., Academic Press, 1976, 255-283.

2. WARE, W. H. The ultimate computer. *IEEE Spectrum* 9, 3 (March 1972), 84-91.

3. WINOGRAD, S. On the time required to perform multiplication, *J. ACM* 14, 4 (1967), 793-802.

4. LAMAGNA, E. A. AND SAVAGE, J. E. Combinational complexity of some monotone functions. *IEEE 15th Annual Symp. on Switching and Automata Theory (1974)*, 140-144.

5. THURBER, K. J. Large scale computer architecture. Parallel and associative processors. Hayden, N. J., 1976.

6. FOSTER, C. C. Content addressable parallel processors, Van Nostrand Reinhold, 1976.

7. KUCK, D. J. A survey of parallel machine organization and programming. *Computing Surveys* 9, 1 (1977), 29-60.

8. MINSKY, M. Form and content in computer science. *J.ACM* 17 (1970), 197-215.

9. CHEN, T. C. Distributed intelligence for user-oriented computing. *AFIPS FJCC Proc.* (1972), 1049-1056.

10. KUCK, D. ET AL. Measurements of parallelism in ordinary FORTRAN programs, *Computer* 1 (1974), 37-46.

11. SHORE, J. E. Second thoughts on parallel processing. *Proc. 1972 IEEE INTERCON*, 358-359.

12. YAMADA, H., AND AMOROSO, S. Tesselation automata. *Inf. and Control* 14 (1969), 299-317.

13. VON NEUMANN, J. A system of 29 states with a general transition rule. *Theory of Self-Reproducing Automata*. A. W. Burks, Ed., Univ. of Illinois Press, 1966, 132-156.

14. ENSLOW, P. H., Jr. Multiprocessors and parallel processing. John Wiley & Sons, N. Y., 1974.

15. FARBER, D. J. A ring network. *Datamation* 21, 2 (Feb. 1975), 44-46.

16. SEARLE, B. C., AND FREBERG, D. E. Microprocessor applications in multiple processor systems. *Computer* 8, 10 (1975), 22-31.

17. TSETLIN, M. L. Automaton theory and modeling of biological systems. Academic Press, N.Y., 1973.

18. HOLLAND, J. H. A universal computer capable of executing an arbitrary number of sub-programs simultaneously. *Proc. AFIPS FJCC* (1959), 108-113.

19. COMFORT, W. T. A modified Holland machine. *Proc. AFIPS FJCC* (1963), 481-488.

20. UNGER, S. H. A computer oriented toward spatial problems. *Proc. IRE* (1958), 1744-1750.

21. SLOTNICK, D. L. Unconventional systems. *Proc. AFIPS SJCC* (1967), 477-481.

22. BARNES, G. ET AL. The ILLIAC IV computer. *IEEE Trans. Computers* 17, 8 (1968), 746-757.

23. SQUIRE., J. S. AND PALAIS, S. M. Programming and design considerations of a highly parallel computer. *Proc. AFIPS SJCC* (1963), 395-400.

24. FOSTER, C. C. Computer architecture, Second Edition. Van Nostrand Reinhold, 1976.

25. MOORE, E. F. Machine Models of self-reproduction. *In Essays on Cellular Automata*, A. W. Burks, Ed., Univ. of Illinois Press, Urbana, 1970, 187-203.

26. MYHILL, J. The converse of Moore's Garden-of-Eden theorem. *Proc. Amer. Math. Soc.* 14 (1963), 685-686.

27. YAMADA, H., AND AMOROSO, S. A completeness problem for pattern generation in tessellation automata. *J. Comput. Syst. Sci.* 4 (1970), 137-176.

28. YAMADA, H., AND AMOROSO, S. Structural and behavioral equivalences of tessellation automata. *Inf. and Control* 18, 1 (1971), 1-31.

29. MARUOKA, A., AND KIMURA, M. Completeness problem of one-dimensional binary scope-3 tessellation automata, *J. Comput. Syst. Sci.* 9, 1 (1974), 31-47.

30. YAKU, T. The constructibility of a configuration in a cellular automaton. *J. Comput. Syst. Sci.* 7, 5 (1973), 481-496.

31. RICHARDSON, D. Tessellations with local transformations. *J. Comput. Syst. Sci.* 6, 5 (1972), 373-388.

32. OSTRAND, T. J. Property preservation by tessellation automata. Ph.D. dissertation, Rutgers Univ., 1972.

33. HERMAN, G. T. and ROZENBERG, G. Developmental systems and languages. American Elsevier, N.Y., 1975.

34. LIEBLEIN, E. A theory of patterns in two-dimensional tessellation space. Ph.D. Thesis, Univ. of Pennsylvania, Penn., 1968.

35. BALZER, R. A. A 8-state minimal solution to the firing squad synchronization problem. *Inf. and Control* 10 (1967), 22-42.

36. SMITH, A. R., III. Real-time language recognition by one-dimensional cellular automata. *J. Comput. Syst. Sci. 6,3* (1972), 233-253.

37. SMITH, A. R., III. Two dimensional formal languages and pattern recognition by cellular automata. *IEEE 12th Annual Symp. Switching and Autom. Theory* (1971), 144-152.

38. WOLPERT, L. The French flag problem: a contribution to the discussion on pattern development and regulation. In C. H. Waddington, Ed., *Towards a Theoretical Biology,* 1, Edinburgh Univ. Press (1968), 125-133.

39. FISCHER, P. C. Generation of primes by a one-dimensional real-time iterative array. *J. ACM* 12, 3 (1965), 388-394.

40. COLE, S. N. Real-time computation by n-dimensional iterative arrays of finite-state machines. *IEEE Trans. Comput.* 18 (1969), 349-365.

41. SEIFERAS, J. I. Observations on nondeterministic multidimensional iterative arrays. *Proc. ACM 6th Annual Symp. on Theory of Computing* (1974), 276-287.

42. WAGNER, E. G. On connecting modules together uniformly to form a modular computer. *IEEE Trans. Electronic Comput.* 15 (1966), 864-873.

43. HARARY, F. Graph theory. Addison-Wesley, Mass., 1969.

44. DORFLER, W. Every regular graph is a quasigroup graph. *Discrete Math.* 10 (1974), 181-183.

45. ORE, O. Studies on directed graphs II. *Annals of Mathem.* 64, 1 (1956), 142-153.

46. NASH-WILLIAMS, C. S. J. A. Infinite graphs--a survey. *J. Comb. Theory* 3 (1967), 286-301.

47. PETERSEN, J. Die theorie der regulären graphen. *Acta mathem.* 15 (1891), 193-220.

48. KUROSH, A. G. Lectures in general algebra. Pergamon Press, 1965.

49. LIU, C. L. Introduction to combinatorial mathematics. McGraw Hill, 1968.

50. MAGNUS, W., KARRAS, A., AND SOLITAR, D. Combinatorial group theory, John Wiley & Sons, 1966.

51. FLYNN, M. J. Some computer organizations and their effectiveness. *IEEE Trans. Comput.* 21, 9 (1972), 948-960.

52. THAYSE, A. Disjunctive and conjunctive operators for Boolean functions. *Phillips Res. Rep. 28*, 1 (1973), 1-16.

53. AHO, A. V., AND ULLMAN, J. D. The theory of parsing, translation, and compiling, Volume 1: Parsing, Prentice-Hall, N.J., 1972.

54. KARP, R. M. The probabilistic analysis of some combinatorial search algorithms. In *Algorithms and Complexity--New Directions and Recent Results*. J. F. Traub, Ed., Academic Press, 1976, 1-20.

55. GROSKY, W. I. On the existence of Garden of Eden configurations. *Information and Control* (1977), to appear.

56. KNUTH, D. E., AND BENDIX, P. B. Simple word problems in universal algebras. In *Computational Problems in Abstract Algebra*. Pergamon Press, 1970, 263-298.

57. LIPOVSKI, G. J. The architecture of a large associative processor. *Proc. AFIPS SJCC* (1970), 385-396.

58. ENSLOW, P. H., Jr. Multiprocessor organization--a survey. *Computing Surveys* 9, 1 (1977), 103-129.

59. WULF, W. A. Minicomputer complexes: progresses and prospects. In *Computer Science and Scientific Computing*, J. M. Ortega, Ed. (1976), 283-306.

60. SAHIN, K. E. Message-based response routing with Selcuk networks. *IEEE Trans. Comput.* 23, 12 (1974), 1250-1256.

61. DAVIES, D. W., AND BARBER, D. L. A. Communication networks for computers. John Wiley & Sons (1973), Chapter 12.

62. AHO, A.V., HOPCROFT, J. E., AND ULLMAN, J. D. The design and analysis of computer algorithms. Addison-Wesley, Mass., 1974.

63. HARTMANIS, J., AND SIMON, J. On the structure of feasible computations. In *Advances in Computers*, M. Rubinoff and M. C. Yovits, Eds., Academic Press, N.Y., 1976, 1-43.

VITA

Renato Ermann was born November 19, 1946, in Santiago, Chile. He received the degree of Civil Electrical Engineer from the University of Chile, Santiago in 1971 and the Master in Electrical Engineering degree from the same institution in 1973. Mr. Ermann earned the Master of Science degree in Information and Computer Science from the Georgia Institute of Technology in 1975.

Mr. Ermann was employed as an assistant professor and research engineer by the Department of Electrical Engineering, University of Chile, Santiago from 1970 to 1973. While at Georgia Tech, he was a Fulbright scholar and a graduate teaching assistant in the School of Information and Computer Science.

Mr. Ermann has three publications and presented one paper on an international conference. He will be joining Bell Laboratories at their Columbus, Ohio location. He is a member of the Institute of Electrical and Electronics Engineers and of the Association for Computing Machinery.